

# **SKRIPSI**

## **PERENCANAAN DAN PEMBUATAN PENDETEKSI DAN PENGHITUNG DENYUT JANTUNG MANUSIA BERBASIS MIKROKONTROLER AT89C2051 DIKOMUNIKASIKAN KE PC MELALUI USB**



**Disusun Oleh :  
DENY HERYADI  
03.17.059**



**JURUSAN TEKNIK ELEKTRO S -1  
KONSENTRASI TEKNIK ELEKTRONIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG**

**MARET 2008**

---

## LEMBAR PERSETUJUAN

### PERENCANAAN DAN PEMBUATAN PENDETEKSI DAN PENGHITUNG DENYUT JANTUNG MANUSIA BERBASIS MIKROKONTROLER AT89C2051 DIKOMUNIKASIKAN KE PC MELALUI USB

## SKRIPSI

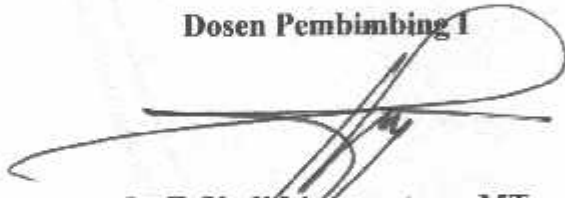
*Disusun Dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Teknik Elektronika Strata Satu (S-1)*

Disusun Oleh :


**DENY HERYADI**  
03.17.059

Diperiksa dan Disetujui

Dosen Pembimbing I

  
**Ir. F. Yudi Limpraptono, MT**  
NIP.V. 1039500274

Dosen Pembimbing II

  
**Joseph Dedy Irawan, ST, MT**  
NIP.132315178

Mengetahui



Ketua Jurusan Teknik Elektro S-1

  
**Ir. F. Yudi Limpraptono, MT**  
NIP.Y. 1039500274

**JURUSAN TEKNIK ELEKTRO S -1  
KONSENTRASI TEKNIK ELEKTRONIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG**

**MARET 2008**



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK ELEKTRONIKA

---

**BERITA ACARA UJIAN SKRIPSI**  
**FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Deny Heryadi  
NIM : 03.17.059  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul Skripsi : Perencanaan dan Pembuatan Pendeteksi dan Penghitung Denyut Jantung Manusia Berbasis Mikrokontroler AT89C2051 Dikomunikasikan ke PC Melalui USB

Dipertahankan di hadapan majelis penguji Skripsi jenjang Strata satu ( S-1 ) pada :

Hari : Senin  
Tanggal : 17 Maret 2008  
Dengan Nilai : 80,4 (A) *84*



**(Ir. Mochtar Asroni, MSME)**  
NIP.Y.1018100036

**Sekretaris Majelis Penguji**

**(Ir. F. Yudi Limpraptono, MT)**  
NIP.Y.1039500274

**Penguji I**

**(Ir. Teguh Herbasuki, MT)**  
NIP.Y.1038900209

**Penguji II**

**(Komang Somawirata, ST, MT)**  
NIP.P.1030100361

## ABSTRAKSI

### PERENCANAAN DAN PEMBUATAN PENDETEKSI DAN PENGHITUNG DENYUT JANTUNG MANUSIA BERBASIS MIKROKONTROLER AT89C2051 DIKOMUNIKASIKAN KE PC MELALUI USB

(Deny Heryadi, 03.17.059, Teknik Elektro S-1/Elektronika S-1)

(Dosen Pembimbing I : Ir. F. Yudi Limpraptono, MT)

(Dosen Pembimbing II : Joseph Dedy Irawan, ST, MT)

Kata kunci : FT232 USB Serial, Mikrokontroler AT89C2051, Denyut Jantung

Jantung merupakan inti dari kehidupan manusia. Jantung menyuplai kebutuhan seluruh tubuh. Gangguan pada jantung akan berdampak pada semua kerja organ tubuh. Cara yang paling mudah untuk mengetahui adanya gangguan pada jantung dengan menghitung jumlah denyut per menit. Secara umum dapat dikelompokkan empat kategori. Pertama, bayi yang baru lahir memiliki denyut jantung per menit normal berkisar 120 - 160 denyut per menit. Kedua, anak usia 1 - 10 (80-120). Ketiga, anak usia diatas 10 tahun dan orang dewasa (60 - 80). Terakhir untuk atlet terlatih berkisar (40 - 60).

Metode yang digunakan pada skripsi ini menggunakan detektor suara dengan menggunakan stetoskop akustik dan metode *non-invasive* menggunakan plesthymograf yang kemudian dikuatkan oleh rangkaian pengkondisi sinyal dan diproses oleh mikrokontroler AT89C2051 dan PC.

Dari hasil pengujian alat didapatkan perbedaan dibandingkan menggunakan alat *pulse oximeter* BioSys BPM 200 dengan nilai *error* keseluruhan pengujian sebesar 8.97%.

## KATA PENGANTAR

Atas Rahmat Tuhan Pengatur Kehidupan, penulis dapat menyelesaikan laporan skripsi dengan judul:

***“Perencanaan dan Pembuatan Pendeteksi dan Penghitung Denyut Jantung Manusia Berbasis Mikrokontroler AT89C2051 Dikomunikasikan ke PC Melalui USB”***

Pembuatan Skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata-1 di Institut Teknologi Nasional Malang. Laporan Skripsi ini merupakan tanggung jawab tertulis atas ilmu pengetahuan yang didapat selama penulis mengikuti kuliah.

Atas terselesaikannya skripsi ini, penulis mengucapkan terima kasih kepada :

- Bapak Prof. Dr. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
- Bapak Ir. Mochtar Asroni, MSME selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
- Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
- Bapak Ir. F. Yudi Limpraptono, MT dan Bapak Joseph Dedy Irawan, ST, MT selaku Dosen Pembimbing yang telah memberikan bimbingan, pengarahan, serta ilmu-ilmu yang sangat berharga sehingga skripsi ini dapat terselesaikan.
- Teman-teman yang telah membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata, penulis mohon maaf kepada semua pihak apabila selama penulisan skripsi ini penulis membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, Maret 2008

Penulis

## DAFTAR ISI

	Halaman
<b>JUDUL</b> .....	i
<b>LEMBAR PERSETUJUAN</b> .....	ii
<b>BERITA ACARA UJIAN SKRIPSI</b> .....	iii
<b>ABSTRAKSI</b> .....	iv
<b>KATA PENGANTAR</b> .....	v
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR GAMBAR</b> .....	xii
<b>DAFTAR TABEL</b> .....	xiv
 <b>BAB I    PENDAHULUAN</b>	
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan .....	3
1.5. Metodologi Penelitian .....	3
 <b>BAB II    LANDASAN TEORI</b>	
2.1. Pendahuluan .....	5
2.2. Cara Kerja Jantung.....	5
2.3. Denyut Jantung Tiap Menit .....	6
2.4. Tranduser .....	7

2.4.1. Transduser Akustik .....	7
2.4.2. Transduser Plesthymograf.....	8
2.5. Rangkaian Pengkondisi Sinyal .....	10
2.6. Mikrokontroler AT89C2051 .....	10
2.6.1. Pendahuluan .....	10
2.6.2. Perangkat Keras Mikrokontroler AT89C2051.....	11
2.6.2.1. Arsitektur AT89C2051 .....	12
2.6.2.2. Konfigurasi Pin .....	13
2.6.2. Organisasi Memori.....	15
2.6.3.1. Program Memori Internal.....	15
2.6.3.2. Data Memory (RAM) Internal .....	15
2.6.3.3. SFR (Register Fungsi Khusus).....	16
2.6.3.4. Sistem <i>Interrupt</i> .....	18
2.6.3.5. <i>Timer/Counter</i> .....	20
2.6.3.6. Metode Pengalamatan.....	22
2.6.3.6.1. Pengalamatan Bit ( <i>Bit Addressing</i> ).....	22
2.6.3.6.2. Pengalamatan Tak Langsung.....	22
2.6.3.6.3. Pengalamatan Berindeks.....	22
2.6.3.6.4. Konstanta <i>Immediate</i> .....	23
2.6.3.7. Metode Pemasukkan Program ke AT89C2051 ...	23
2.6.3.7.1. <i>Internal Address Counter</i> .....	23
2.6.3.7.2. Algoritma <i>Programming</i> .....	23
2.6.3.7.3. Program Verifikasi .....	25
2.6.3.7.1. <i>Chip Erase</i> .....	25



2.6.3.7.1. <i>Reading The Signature Byte</i> .....	25
2.7. Modul FT232 USB Serial <i>Converter</i> .....	26
2.7.1. Kabel USB .....	28
2.7.2. Sinyal USB .....	30
2.7.3. Sinyal Penyambungan dan Pemutusan .....	32

### **BAB III PERANCANGAN DAN PEMBUATAN ALAT**

3.1. Pendahuluan .....	35
3.2. Perancangan Perangkat Keras .....	37
3.2.1. Perancangan Transduser .....	37
3.2.2. Perancangan Rangkaian Pengkondisi Sinyal .....	39
3.2.2.1. Rangkaian Pengkondisi Sinyal Transduser Akustik .....	39
3.2.2.1.1. Rangkaian Penyangga ( <i>Buffer</i> ) .....	39
3.2.2.1.2. Rangkaian Penguat Tak Membalik ( <i>NonInverting Amplifier</i> ).....	40
3.2.2.1.3. Rangkaian Pembanding Tegangan ( <i>Voltage Comparator</i> ).....	41
3.2.2.2. Rangkaian Pengkondisi Sinyal Transduser Plesthymograf .....	42
3.2.2.2.2. Rangkaian <i>Converter (I-V)</i> .....	42
3.2.2.2.2. Rangkaian Penyangga ( <i>Buffer</i> ) .....	44
3.2.2.2.1. Rangkaian Pembanding Tegangan ( <i>Voltage Comparator</i> ) .....	44

3.2.3. Mikrokontroler AT89C2051 .....	45
3.2.4. Modul FT232 USB Serial <i>Converter</i> .....	46
3.2.5. <i>Personal Computer</i> (PC) .....	46
3.3. Perancangan Perangkat Lunak .....	47
3.3.1. Flowchart Keseluruhan Sistem .....	48
3.3.2. <i>Flowchart</i> Program Mikrokontroler .....	49
3.3.3. <i>Flowchart</i> Program di PC .....	50

#### **BAB IV PENGUJIAN ALAT**

4.1. Pendahuluan .....	51
4.2. Tujuan Pengujian .....	51
4.3. Pengujian Modul FT232 USB Serial <i>Converter</i> .....	51
4.4. Pengujian Transduser .....	60
4.5. Pengujian Rangkaian Pengkondisi Sinyal .....	62
4.6. Pengujian Keseluruhan Alat .....	63
4.7. Spesifikasi Alat .....	68

#### **BAB V PENUTUP**

5.1. Kesimpulan .....	70
5.2. Saran .....	71

#### **DAFTAR PUSTAKA**

#### **LAMPIRAN**

## DAFTAR GAMBAR

Gambar 2-1	Stetoskop .....	7
Gambar 2-2	Simbol <i>Microphone</i> .....	8
Gambar 2-3	Simbol LED .....	8
Gambar 2-4	Simbol Fotodioda.....	9
Gambar 2-5	Spektrum Penyerapan Hb dan HbO2 .....	9
Gambar 2-6	Simbol Op Amp .....	10
Gambar 2-7	Blok Diagram Mikrokontroler AT89C2051 .....	12
Gambar 2-8	Konfigurasi Pin AT89C2051 .....	13
Gambar 2-9	Kegunaan <i>Interrupt Enable Register</i> .....	19
Gambar 2-10	Konfigurasi dan Kegunaan TMOD .....	20
Gambar 2-11	<i>Timer/Counter Mode Control Register</i> .....	21
Gambar 2-12	Mode <i>Flash Programming</i> .....	26
Gambar 2-13	Modul FT232 USB Serial <i>Converter</i> .....	28
Gambar 2-14	Alokasi Pin Modul FT232 USB Serial <i>Converter</i> .....	28
Gambar 2-15	Konektor USB .....	29
Gambar 2-16	Konektor Pin USB tipe A .....	30
Gambar 2-17a	Perkabelan USB Kecepatan Rendah .....	32
Gambar 2-17b	Perkabelan USB Kecepatan Penuh .....	32
Gambar 2-18	Sebuah Piranti Dilepas dari Port USB .....	33
Gambar 2-19	Piranti USB Kecepatan Tinggi Dihubungkan ke port USB .....	33
Gambar 2-20	Piranti USB Kecepatan Rendah Dihubungkan ke port USB....	34
Gambar 3-1	Blok Diagram Sistem .....	35

Gambar 3-2	Tranduser Akustik .....	37
Gambar 3-3	Rangkaian <i>Microphone</i> .....	38
Gambar 3-4	Plesthymograf.....	38
Gambar 3-5	Rangkaian LED dan Fotodioda .....	39
Gambar 3-6	Rangkaian Penyangga ( <i>Buffer</i> ).....	40
Gambar 3-7	Rangkaian Penguat Tak Membalik ( <i>Non-Inverting Amplifier</i> )	41
Gambar 3-8	Rangkaian Pembanding Tegangan (Komparator) .....	42
Gambar 3-9	Rangkaian <i>Converter (I-V)</i> .....	43
Gambar 3-10	Rangkaian Penyangga ( <i>Buffer</i> ).....	44
Gambar 3-11	Rangkaian Pembanding Tegangan.....	45
Gambar 3-12	Alokasi pin mikrokontroler AT89C2051 .....	45
Gambar 3-13	Rangkaian Pin Modul FT232 USB Serial Converter.....	46
Gambar 3-14	Perencanaan Tampilan pada PC .....	47
Gambar 4-1	Blok Pengujian Modul USB .....	52
Gambar 4-2	PC Mendeteksi <i>Hardware</i> baru .....	53
Gambar 4-3	Perintah Untuk Memilih Cara Peng- <i>install</i> -an .....	53
Gambar 4-4	Perintah Memberitahu Lokasi Driver .....	54
Gambar 4-5	Lokasi Driver .....	54
Gambar 4-6	Proses <i>Transfer</i> Driver USB.....	55
Gambar 4-7	Proses Peng- <i>install</i> -an Selesai .....	55
Gambar 4-8	Tampilan Delphi Saat <i>Setting</i> Komponen Vacomm .....	59
Gambar 4-9	Tampilan saat Program Dijalankan .....	59
Gambar 4-10	Nyala Indikator pada Modul FT232 USB Serial <i>Converter</i> ....	60
Gambar 4-11	Blok Pengujian Tranduser .....	60

Gambar 4-12	Sinyal Keluaran Tranduser Akustik .....	61
Gambar 4-13	Sinyal Keluaran Penguat Tranduser Akustik .....	61
Gambar 4-14	Sinyal Keluaran Tranduser Plesthymograf .....	61
Gambar 4-15	Sinyal Keluaran Penguat Tranduser Plesthymograf .....	62
Gambar 4-16	Blok Pengujian Rangkaian Pengkondisi Sinyal .....	62
Gambar 4-17	Sinyal Keluaran Komparator Tranduser Akustik .....	63
Gambar 4-18	Sinyal Keluaran Komparator Tranduser Plesthyomgraf .....	63
Gambar 4-19	Blok Pengujian Pendeteksi dan Penghitung Denyut Jantung...	64
Gambar 4-20	Tampilan Awal Menjalankan Program Delphi.....	65
Gambar 4-21	Tampilan Pilihan Kategori pada Program Delphi .....	66
Gambar 4-22	Tampilan Hasil Pengukuran .....	66
Gambar 4-23	Foto Keseluruhan Alat .....	69
Gambar 4-24	BioSys BPM 200 .....	69

## DAFTAR TABEL

Tabel 2-1	Frekuensi Denyut Jantung Per Menit ( <i>beat per minute</i> ).....	6
Tabel 2-2	Pengaturan RS0-RS1 Untuk <i>Select Register Bank</i> .....	16
Tabel 2-3	<i>Special Function Register</i> .....	16
Tabel 2-4	Alamat Sumber Interupsi .....	18
Tabel 2-5	Mode Operasi <i>Timer/Counter</i> .....	21
Tabel 4-1	Hasil Pengujian dengan Tranduser Plesthymograf .....	67
Tabel 4-2	Hasil Pengujian dengan Tranduser Akustik .....	67

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Teknologi *interface* (antarmuka) yang mengkomunikasikan sebuah PC dengan devais lain sudah sangat berkembang. Baik yang masih menggunakan kabel maupun *wireless* (tanpa kabel). Saat ini antar muka yang paling sering dipakai oleh hampir semua perangkat dan aksesoris komputer adalah Universal Serial Bus (USB). Dengan dimensinya yang ringkas dan *user friendly*, USB menjadi antar muka yang paling banyak yang dimiliki dan digunakan dari sebuah komputer (PC). Karena itu USB dapat dioptimalkan pemakaiannya untuk berkomunikasi secara serial, misalnya dengan mikrokontroler yang diimplementasikan untuk suatu aplikasi.

Modul detektor denyut jantung manusia untuk menghitung banyaknya denyut jantung dalam setiap menit yang berbasiskan sebuah mikrokontroler menerima data analog dari tranduser yang kemudian dikonversi menjadi sinyal yang disesuaikan dengan level tegangan digital menggunakan rangkaian pengkondisi sinyal sehingga dapat diproses oleh mikrokontroler. Mikrokontroler dikomunikasikan dengan sebuah PC dapat dilakukan dengan cara serial dan paralel. Keluaran dari mikrokontroler dapat dihubungkan ke port komunikasi pada PC. Dengan demikian data yang didapat dari detektor dapat dikirim ke PC.

Dengan menggunakan modul FT232 Serial-USB *Converter* sebuah mikrokontroler dapat dikomunikasikan dengan sebuah PC melalui USB secara serial. Modul FT232 Serial-USB *Converter* adalah modul yang berbasiskan chip

FT232 *Serial-USB Converter* yang secara khusus digunakan untuk mengkonversikan pengiriman data serial ke pengiriman data serial USB.

### 1.2. Rumusan Masalah

Dalam perencanaan dan pembuatan pendeteksi dan penghitung denyut jantung manusia berbasis mikrokontroler AT89C2051 yang dikomunikasikan dengan PC melalui USB dapat dirumuskan beberapa masalah sebagai berikut :

1. Bagaimana membuat suatu rangkaian transduser dan penguat yang dapat membaca denyut nadi secara kontinyu.
2. Bagaimana mengkomunikasikan suatu aplikasi berbasis mikrokontroler ke sebuah *Personal Computer* (PC) melalui antarmuka *Universal Serial Bus* (USB) .

### 1.3. Batasan Masalah

Agar pembahasan dari perencanaan dan pembuatan pendeteksi dan penghitung denyut jantung manusia berbasis mikrokontroler AT89C2051 yang dikomunikasikan dengan PC melalui USB tidak terlalu meluas maka penyusun perlu membuat batasan-batasan masalah yang meliputi :

1. Menggunakan modul FT232 *USB-Serial Converter* sebagai komunikasi mikrokontroler dengan PC
  2. Menggunakan mikrokontroler AT89C2051
  3. Tidak menampilkan bentuk karakteristik sinyal jantung, hanya menampilkan indikator denyut yang terukur.
-



4. pengukuran dilakukan dalam kondisi istirahat, dalam suasana tenang dan di dalam ruangan.

#### **1.4. Tujuan**

Tujuan dari penulisan skripsi ini adalah untuk membuat suatu alat penghitung denyut jantung manusia untuk mengetahui jumlah denyut jantung per menit seseorang, dengan menggunakan modul FT232 Serial-USB *Converter* dan Mikrokontroler AT89C2051 sebagai kendali utama.

#### **1.5. Metodologi Penelitian**

Untuk merealisasikan skripsi diatas, maka metodologi penulisan yang digunakan adalah sebagai berikut :

##### **1. KAJIAN PUSTAKA**

Bertujuan untuk mengumpulkan literatur berisikan teori yang berhubungan dengan perencanaan alat, Dipadukan dengan teori yang didapat dibangku kuliah

##### **2. PERENCANAAN DAN PEMBUATAN ALAT**

Bertujuan untuk membuat diagram blok rangkaian yang sesuai dengan rencana kerja, kemudian direalisasikan dengan melaksanakan perencanaan dan pembuatan alat berdasarkan diagram blok rangkaian yang disusun.

---

### 3. STUDI ANALISA ALAT

Bertujuan untuk melakukan analisa pengujian alat yang telah dirancang, apakah telah sesuai dengan fungsi kerja yang diharapkan atau belum.

### 4. PENYUSUNAN BUKU LAPORAN

Bertujuan untuk menyusun data laporan berpedoman pada alat yang selesai dibuat beserta kesimpulan cara kerja dari alat tersebut.

---

## BAB II

### LANDASAN TEORI

#### 2.1. Pendahuluan

Landasan teori sangat membantu untuk dapat memahami suatu sistem. Landasan teori digunakan juga sebagai acuan di dalam merencanakan suatu sistem. Dengan pertimbangan diatas maka landasan teori merupakan bagian yang harus dipahami untuk pembahasan lebih lanjut. Dalam landasan teori ini akan dibahas teori dasar yang berhubungan dengan cara kerja jantung, denyut jantung tiap menit, tranduser, pengkondisi sinyal, Mikrokontroler *AT89C2051* dan *FT232 Serial-USB Converter*.

#### 2.2. Cara Kerja Jantung

Fungsi jantung secara umum adalah memompa darah yang mengandung oksigen ( $O_2$ ) ke seluruh sistem sirkulasi. Sistem sirkulasi pada jantung manusia terdiri atas tiga saluran yang saling berkaitan antara satu dengan yang lainnya dan jika salah satu mengalami gangguan maka akan menyebabkan kelainan jantung.

Sistem sirkulasi ini terbentuk atas saluran-saluran yang meliputi :

- Pembuluh nadi (arteri)
- Pembuluh balik (vena)
- Pembuluh kapiler

Secara fisis, jantung terdiri atas 4 ruangan yaitu 2 *atrium* (serambi) kanan dan kiri serta 2 *ventrikel* (bilik) kanan dan kiri. Fungsi utama *ventrikel* adalah memompa darah dan hasilnya disimpan oleh *atrium*. Pada saat pemompaan darah,

dinding *ventrikel* berkontraksi yang dikenal dengan istilah *Sistole* diikuti oleh proses penyimpanan darah yang dikenal dengan istilah *Diastole* dilanjutkan dengan istirahat (relaksasi) dan seterusnya.

**2.3. Denyut Jantung Tiap Menit**

Keterangan frekuensi denyut jantung per menit dapat dilihat pada tabel berikut.

**Tabel 2.1**

**Frekuensi Denyut Jantung Per Menit (*beat per minute*)**

Denyut Jantung Per Menit	Keterangan
120-160 bpm	Bayi Baru Lahir
80-120 bpm	Anak Usia 1-10 tahun
60-80 bpm	Anak Usia > 10 Tahun dan Dewasa
40-60 bpm	Atlet Terlatih

Jumlah denyut jantung tiap menit sangat membantu diagnosa kondisi baik fisik maupun mental seseorang. Perubahan kondisi fisik dan mental seseorang tidak selalu tampak. Perubahan ini berdampak langsung terhadap perubahan kecepatan irama denyut jantung tiap menitnya. Kecepatan denyut seseorang dari satu menit ke menit yang lain hampir selalu berubah. Perubahan ini dipengaruhi oleh kondisi mental, fisik, temperatur sekitar, usia dan jenis kelamin.

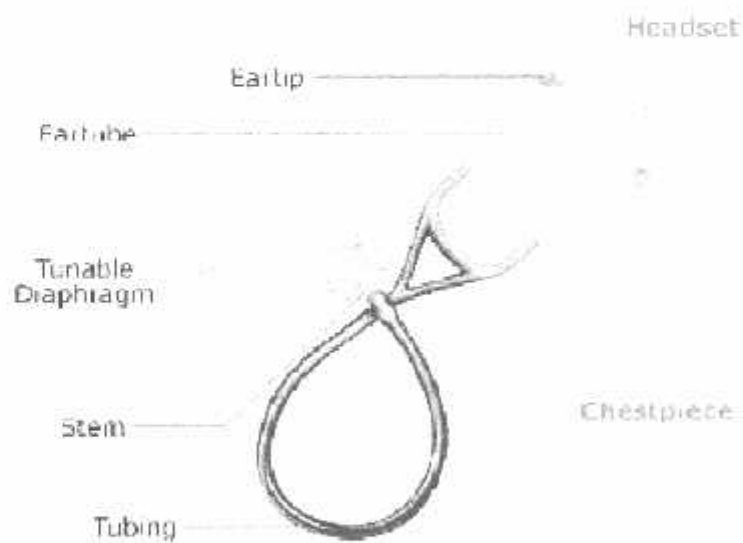
Sinus Bradikardia (kurang dari 60 *bpm*), Sinus Takikardia (lebih dari 100 *bpm*), Takikardia Abnormal (40-250 *bpm*), Flutter (250-350 *bpm*) dan Fibrilasi

(lebih dari 350 *bpm*) merupakan suatu bentuk gangguan pada irama atau frekuensi denyut jantung tiap menitnya. Gangguan ini disebut aritmia.

## 2.4. Transduser

### 2.4.1. Transduser Akustik

Transduser ini digunakan sebagai detektor suara detak jantung. Transduser ini terdiri dari stetoskop yang digunakan untuk menerima suara detak jantung dari dinding dada manusia ke pendengaran kemudian disambungkan ke *microphone condenser*. Bagian stetoskop yang menempel ke dinding dada ada dua macam, diafragma dan bell. Diafragma paling sesuai untuk mendengarkan suara detak jantung. Bagian ini harus terpasang pada dinding dada dengan tekanan cukup sehingga bila diangkat tampak bekas dikulit. Sedangkan bell dipasang pada dinding dada dengan tekanan ringan.



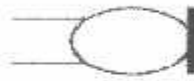
Gambar 2.1

Stetoskop<sup>[5]</sup>

Bagian-bagian utama stetoskop:

- *Tunable Diaphragm* : bagian yang menempel ke dinding dada untuk menerima suara. Terdiri dari dua bagian, diafragma dan *bell*.
- Pipa : untuk melewatkan suara menuju pendengaran manusia.
- *Eartip* : bagian yang dipasang di pendengaran manusia.

*Microphone* adalah sebuah transduser yang mengubah sinyal suara menjadi sinyal-sinyal listrik sesuai pola suara yang diterima. Sistem elektronik akan merespon sinyal-sinyal suara menggunakan komponen ini baik untuk direkam ataupun dikenali.



Gambar 2.2

Simbol *Microphone*<sup>[6]</sup>

#### 2.4.2. Tranduser Plesthymograf

Plesthymograf yang digunakan pada alat ini adalah *photo-plesthymograph*. Plesthymograf biasa digunakan untuk pengukuran tekanan darah dengan metode *non-invasive*. Tranduser ini terdiri dari LED (*Light Emitting Diode*) yang memancarkan spectrum cahaya merah dan spektrum inframerah. Jadi dalam satu LED mamancarkan dua spektrum dengan panjang gelombang yang berbeda, sebagai detektornya digunakan *photodiode*. Fotodioda adalah sebuah komponen yang peka terhadap cahaya dan menghasilkan perubahan tegangan berdasarkan intensitas spektrum yang diterima.



Gambar 2.3

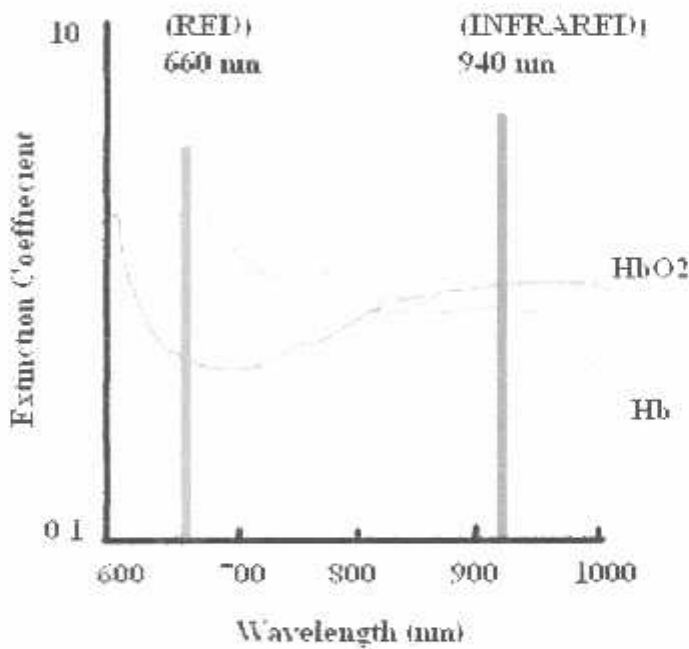
Simbol LED<sup>[6]</sup>



Gambar 2.4

Simbol fotodioda<sup>[8]</sup>

Perubahan tegangan fotodioda akan mengalami fluktuasi (naik turun) seiring aliran darah pada jari yang dikenai cahaya dari LED. Perubahan itulah yang mengindikasikan adanya denyut pada aliran darah yang dipompa oleh jantung. Perbedaan dalam spektrum penyerapan HbO<sub>2</sub> dan Hb digunakan untuk pengukuran saturasi oksigen arteri (jaringan dan pigmentasi menyerap spektrum biru, hijau dan kuning dan air menyerap panjang gelombang inframerah).



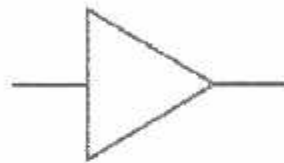
Gambar 2.5

Spektrum Penyerapan Hb dan HbO<sub>2</sub><sup>[7]</sup>

## 2.5. Rangkaian Pengkondisi Sinyal

Sinyal keluaran dari transduser belum sesuai dengan level tegangan yang dapat diproses oleh mikrokontroler. Sehingga untuk dapat diproses oleh mikrokontroler maka sinyal ini harus disesuaikan dengan level tegangan TTL.

Untuk menyesuaikan tegangan keluaran transduser ke level tegangan mikrokontroler maka diperlukan rangkaian pengkondisi sinyal yang merupakan sebuah rangkaian penguat operasional (op amp, *operational amplifier*). Sinyal keluaran transduser dikondisikan sedemikian rupa oleh rangkaian ini sehingga dapat sesuai dengan level tegangan mikrokontroler (TTL).



Gambar 2.6

Simbol Op Amp<sup>[6]</sup>

Rangkaian pengkondisi sinyal disini menggunakan rangkaian penyangga (*buffer*), rangkaian penguat membalik (*inverting amplifier*), rangkaian penguat tak membalik (*non-inverting amplifier*) dan rangkaian pembanding tegangan (*voltage comparator*).

## 2.6. Mikrokontroler AT89C2051

### 2.6.1. Pendahuluan

Perbedaan mendasar antara mikrokontroler dengan mikroprosesor adalah mikrokontroler selain memiliki CPU juga dilengkapi memori dan input output yang merupakan kelengkapan dari sistem minimum mikrokomputer, sehingga



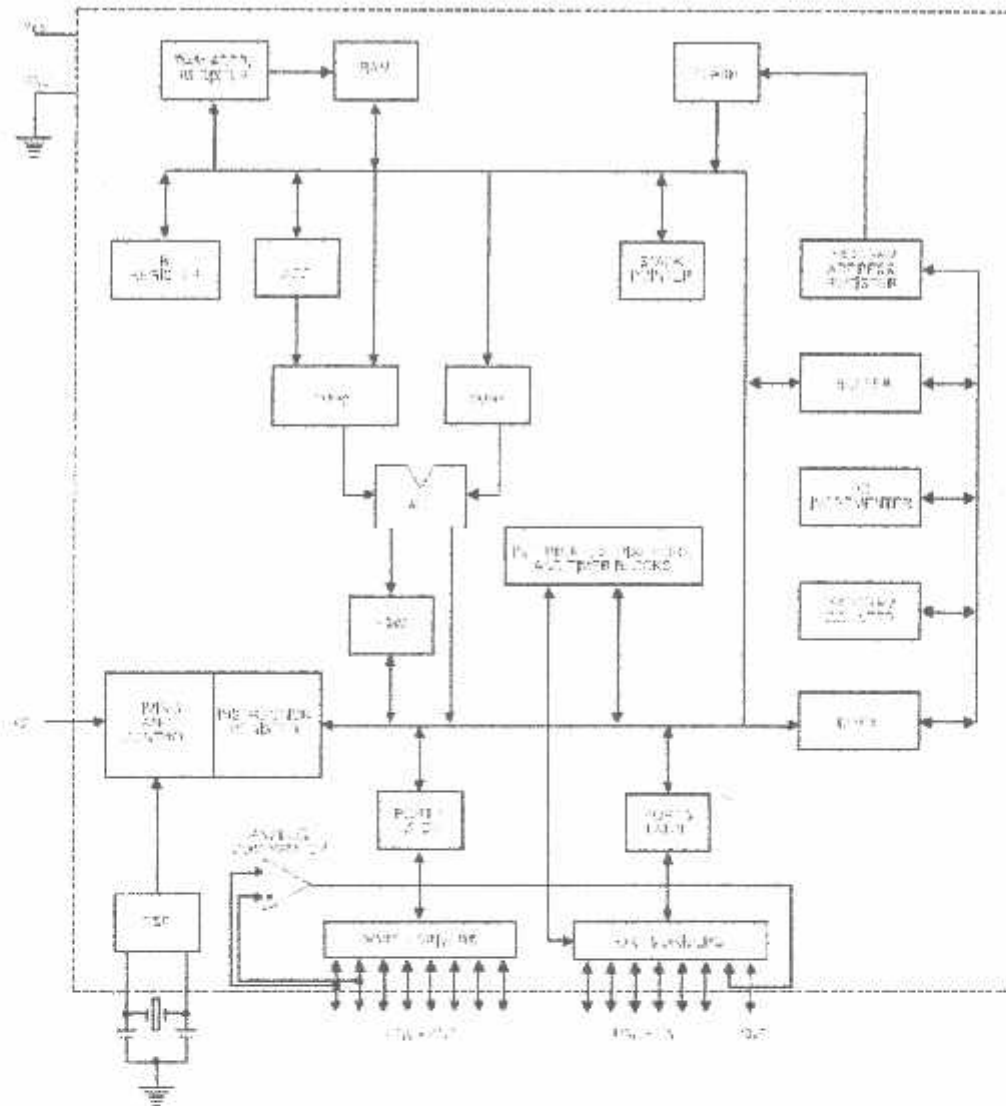
sebuah mikrokontroler dapat dikatakan sebagai mikrokomputer dalam keping tunggal ( *Single Chip Microcomputer* ) yang dapat berdiri sendiri.

Mikrokontroler AT89C2051 adalah keluarga MCS-51, membutuhkan daya yang rendah, memiliki performa yang tinggi, dan merupakan mikrokomputer 8 bit yang dilengkapi 2 Kbyte EPROM ( *Erasable and Programmable Read Only Memory* ) dan 128 RAM byte internal. Program memory dapat diprogram ulang dalam sistem atau dengan menggunakan alat yang dibuat dalam skripsi ini.

Dalam mikrokontroler terdapat dua hal yang mendasar yaitu perangkat keras dan perangkat lunak yang keduanya saling terkait dan mendukung.

#### **2.6.2. Perangkat Keras Mikrokontroler AT89C2051**

- CPU 8bit
  - *Memory*
  - Port input output yang dapat diprogram
  - *Timer dan counter*
  - Sumber *interrupt*
  - Port serial yang dapat diprogram
  - *Oscillator dan Clock*
-



**Gambar 2.7**

### Blok Diagram Mikrokontroler AT89C2051<sup>[3]</sup>

#### 2.6.2.1. Arsitektur AT89C2051

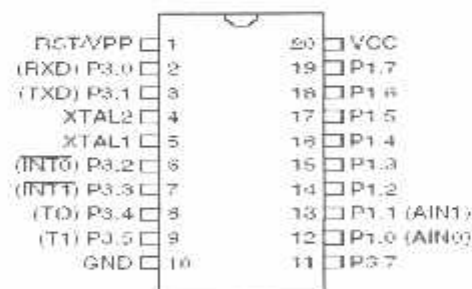
Arsitektur AT 89C2051 adalah sebagai berikut:

1. CPU ( *Central Processing Unit* ) 8 bit dengan register A (*Accumulator*) dan B
2. 16 Bit *Program Counter* (PC) dan *Data Pointer* (DPTR)
3. 8 bit *Program Status Word* (PSW)

4. 8 bit *Stack Pointer*
5. 2Kbyte untuk AT89C2051 internal EPROM
6. 128 byte internal RAM
  - 4 bank register, masing – masing berisi 8 register
  - 16 byte yang dapat dialamati pada bit level
  - 80 byte *general pupose memory data*
7. 16 pin input output yang tersusun atas P1 dan P3, masing – masing 8 bit
8. 2 buah *timer* dan *counter*
9. *Receiver / Transmitter* dan serial *full duplex* : SBUF
10. *Control Register*, yaitu ; TCON, TMOD, SCON, PCON, IP dan IE
11. 5 buah sumber *interrupt* (2 buah sumber interupt internal dan 3 buah sumber interupt external )
12. *Oscillator* dan *Clock Internal*

#### 2.6.2.2. Konfigurasi Pin AT89C2051

Mikrokontroler AT89C2051 terdiri atas 20 pin, dengan konfigurasi sebagai berikut :



**Gambar 2.8**

#### Konfigurasi Pin AT89C2051<sup>[3]</sup>

Fungsi dari tiap – tiap pin adalah sebagai berikut:

1. Vcc (Supply Tegangan)
  2. GND (Ground)
  3. Port 1 adalah port I/O bidireksional 8 bit. Pin port P1.2 sampai P1.7 memberikan tarikan internal. P1.1 berperan sebagai input positif (AIN0) dan input negatif (AIN1) dari komparator analog presisi on chip. Buffer output p1 bisa diturunkan sampai 20 mA dan bisa mengarahkan diplay LED secara langsung. Jika 1 untuk pin port 1, maka dia bisa digunakan sebagai input. Jika P1.2 sampai P1.7 digunakan sebagai input digunakan secara eksternal, maka ia akan dijadikan sebagai arus sumber (I<sub>LE</sub>) karena tarikan internal. Port 1 juga menerima kode selama pemrograman dan pembuktian Flash
  4. Port 3 memiliki P3.0 sampai P3.5, P3.7 dan port 3 terdiri dari tujuh pin I/O bidireksional dengan menarik internal. P3.6 memiliki ikatan keras sebagai suatu input untuk output komparator on chip dan tidak bisa dijangkau sebagai pin I/O tujuan umum. Buffer output port 3 bisa diturunkan sampai 20mA. Jika ditulis untuk pin port 3, maka pin bisa ditarik ke atas engan penarikan dan bisa digunakan sebagai input. Sebagai input, pin port 3 yang ditarik ke bawah secara eksternal akan menjadikan arus sumber (I<sub>in</sub>) karena tarikan ke atas. Port 3 berperan sebagai fungsi fitur khusus dari AT 89C2051, port 3 juga menerima sinyal pengendalian dari pemrograman dan penjelasan flash.
  5. RST adalah input reset.
  6. X-TAL1 dan X-TAL2 . Pin ini dihubungkan dengan kristal bila menggunakan oscillator eksternal. X-TAL1 merupakan input *inverting*
-

*oscillator amplifier* sedangkan X-TAL2 merupakan output *inverting oscillator amplifier*.

### 2.6.3. Organisasi Memori

Di dalam AT89C2051 ruang alamat telah dibedakan untuk program memori dan memori data.

#### 2.6.3.1. Program memori internal

- AT89C2051 memiliki pemrograman memori sebesar 2Kbyte dengan ruang alamat 0000H-07FFH. Jika alamat alamat program lebih tinggi dari pada 07FFh, yang melebihi kapasitas ROM internal menyebabkan AT89C2051 secara otomatis mengambil *code byte* dari memori eksternal. *Code byte* juga dapat diambil hanya dari *external memory* dengan alamat 0000H-07FFH dengan cara menghubungkan pin ke ground.

#### 2.6.3.2. Data Memory (RAM) Internal

Ruang alamat bawah memori data (RAM) internal dengan kapasitas 128 byte yaitu 00H-07H yang terbagi atas 3 daerah, yaitu:

- Empat register bank

Setiap bank terdiri dari 8 register (R0-R7) sehingga jumlah register untuk keempat bank register (bank 0- bank 3) menjadi 32 buah register yang menempati ruang alamat 00H-1FH. Mengaktifkan salah satu *bank register* dapat dilakukan dengan mengatur RS0- RS1 pada PSW (*Program Status Word*).

- *Bit Addressable*

Terdiri atas 16 *byte* yang berada pada alamat 20H-2FH. Masing masing 128 lokasi bit ini dapat dialamati secara langsung.

▪ *Strach Pad Area*

Terdiri atas 80 *byte* yang secara langsung dan digunakan untuk keperluan umum (*general purpose*) misalnya digunakan untuk lokasi *stack*.

Tabel 2.2

Pengaturan RS0-RS1 Untuk *Select Register Bank*

RS1	RS0	Select Regiter Bank
0	1	Bank0
0	1	Bank1
1	0	Bank2
1	1	Bank3

2.6.3.3. SFR (Register Fungsi Khusus)

Suatu pin *area on chip* yang disebut dengan fungsi khusus ditunjukkan tabel berikut ini

Tabel 2.3

*Special Function Register*

0F8H								0FH
0F0H	B 00000000							0F7H
0E8H								0EFH

0E0H	ACC							0E7H
	00000000							
0D8H								0DFH
0D0H	PSW							0D7H
	00000000							
0C8H								0CFH
0C0H								0C7H
0B8H	IP							0B7H
	XXX00000							
0B0H	P3							0B7H
	11111111							
0A8H	0XX00000							0AFH
0A0H								0A7H
98H	SCON	SBUF						9FH
	00000000	XXXX						
		XXXX						
90H	P1							97H
	11111111							
88H	TCON	TMOD	TL0	TL1	TH0	TH1		8FH
	00000000	000000	00000000	00000000	00000000	00000000		
		00						
80H		SP	DPL	DPH			PCON	87H
		000011	00000000	00000000			0XXX000	
							0	

Tidak semua pengalamatan digunakan, dan semua alamat yang digunakan tidak semua diterapkan dalam *chip*. Akses baca pada alamat ini umumnya akan mengembalikan detik acak, dan akses tulis tidak memiliki pengaruh yang tidak ditentukan.

*Software user* hendaknya tidak menulis satu untuk lokasi yang tidak terdaftar, karena mungkin digunakan untuk produk masa mendatang untuk mendukung fitur baru. Dalam hal ini nilai reset atau tidak aktif dari bit baru ini akan selalu menjadi nol.

#### 2.6.3.4. Sistem *Interrupt*

Mikrokontroler AT89C2051 memiliki 5 sumber *interrupt* yang dapat membangkitkan permintaan yaitu : INT0, INT1, T0, T1, dan *port* serial.

Saat terjadinya *interrupt* mikrokontroler secara otomatis akan menuju subrutin pada alamat tersebut. Setelah *interrupt service* selesai dikerjakan, mikrokontroler akan mengerjakan program semula. Dua sumber eksternal adalah INT0 dan INT1 dimana kedua *interrupt* eksternal akan aktif atau transisi tergantung isi IT0 dan IT1 pada register TCON. *Interrupt* T0 dan T1 aktif pada saat timer yang sesuai mengalami *rollover*. Interupsi serial dilakukan dengan melakukan oprasi OR pada R1 dan T1. Tiap – tiap sumber interupsi dapat *enable* atau *disable* secara *software*.

Tingkat prioritas semua sumber interupsi dapat diprogram sendiri – sendiri dengan set atau *clear* bit pada SFRS (*Interrupt Priority*).

**TABEL 2.4**

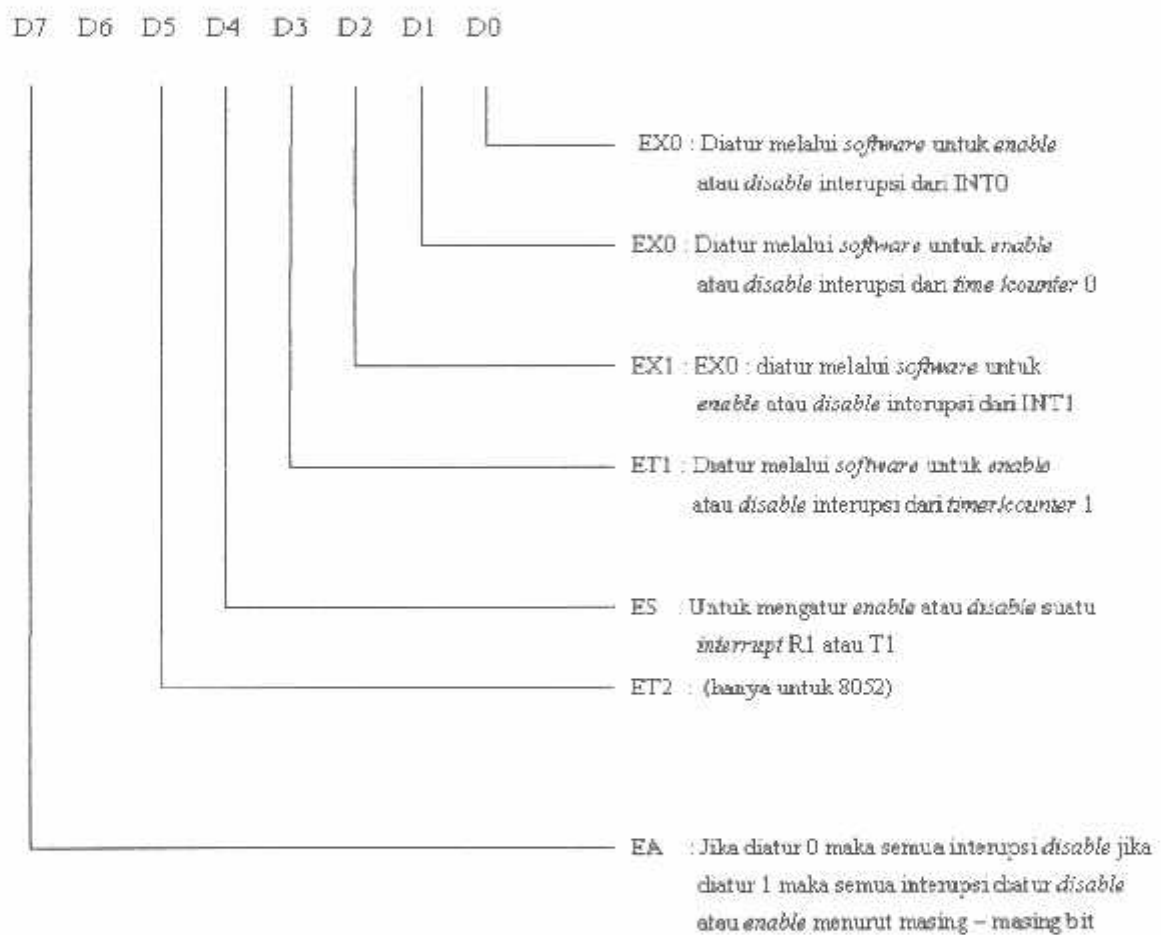
#### Alamat Sumber Iterupsi

Sumber <i>Interrupt</i>	Alamat Awal
Interupsi Luar 0 (INT0)	0003H
Pewaktu / pencacah 0 (T0)	003BH



Interupsi Luar 1 (INT1)	001BH
Pewaktu / pencacah 1 (T1)	001BH
Port Serial	0023H

Register yang berperan dalam mengatur aktif tidaknya interupsi adalah *input enable* register, berikut adalah susunan dari bit – bit beserta kegunaannya



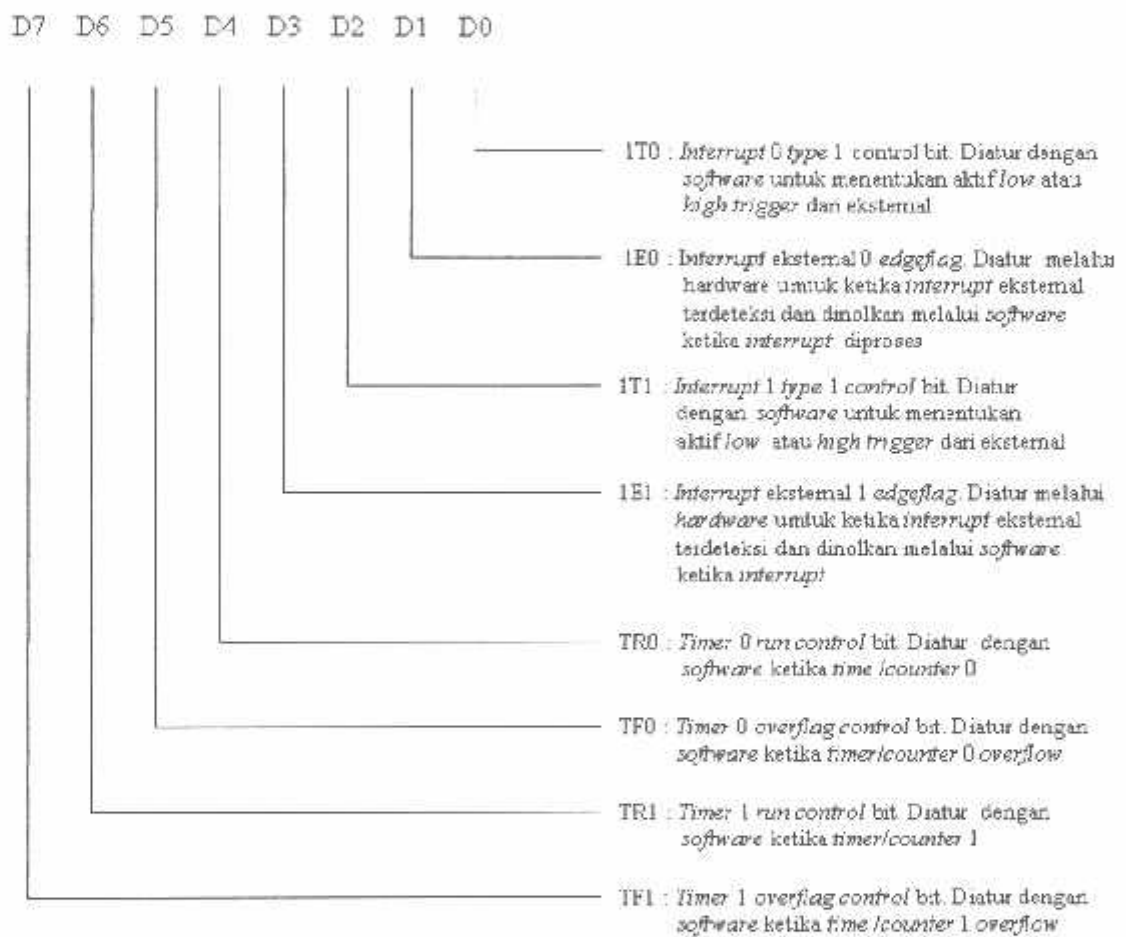
**Gambar 2.9**

**Kegunaan Interrupt Enable Register**

### 2.6.3.5. Timer / Counter

Pengendalian kerja dari *timer/counter* dilakukan dengan pengaturan register yang berhubungan dengan kinerja dari *timer/counter* yaitu melalui sebuah *timer/counter mode control*.

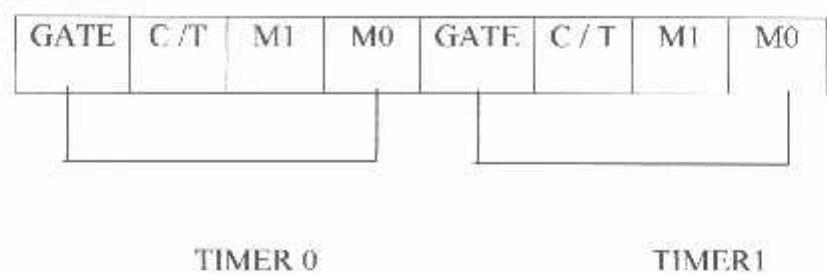
Untuk mengaktifkan *timer/counter* yang meliputi penentuan fungsi sebagai *timer* atau sebagai *counter* serta pemilihan mode operasi dapat diatur melalui TMOD yang beralamat pada 98H. Konfigurasi yang dimaksud adalah sebagai berikut :



Gambar 2.10

### Konfigurasi Dan Kegunaan TMOD

Tmod : *Timer/Counter Mode Control Register*



Gambar 2.11

*Timer/Counter Mode Control Register*

Tabel 2.5

*Mode Operasi Timer/Counter*

M1	M0	Oprating Mode
0	0	Timer 13 bit
0	1	Timer / Counter 16 bit
1	0	16 bir auto reload Timer / Counter
1	1	TLO dari timer adalah 8 bit timer / counter dikendalikan oleh kontrol bit timer 0. TH0 adalah timer 8 bit yang dikendalikan oleh timer 1 control bit

▪ *Gate*

Bila *gate* = 1, *Timer/Counter* x *enable* hanya pada saat pin INTx tinggi dari Trx 1. Saat *gate* 0, *timer/counter enable* jika bit Trx 1.

- *CT*

Jika bit  $C/T = 0$ , maka *timer/counter* x akan berfungsi sebagai *timer*. Jika  $C/T = 1$ , maka *counter* x akan berfungsi sebagai *counter*.

- *M1 dan M2*

Menentukan mode

#### 2.6.3.6. Metode pengalamatan

##### 2.6.3.6.1. Pengalamatan Bit (*Bit Addressing*)

Pengalamatan langsung tiap bit ini hanya dilakukan pada lokasi RAM internal yaitu 20H-2FH, dan sebagian SFR, port 1, port 3, TCON register, SCON register, IE register, PSW register, ACC DAN b register.

##### 2.6.3.6.2. Pengalamatan Tak Langsung

Pada pengalamatan tak langsung, instruksi menunjukkan suatu register yang isinya adalah alamat dari *operand*, eksternal dan internal RAM dapat dialamati secara tak langsung. Register alamat untuk data dengan lebar 8 bit dapat berupa R0 dan R1 yang digunakan untuk memilih register bank atau *stack pointer*. Register alamat untuk data dengan lebar 16 bit digunakan untuk data *pointer* (DPTR).

##### 2.6.3.6.3. Pengalamatan Berindeks

Yang dapat diakses dengan pengalamatan berindeks hanya memori program. Mode ini dimaksudkan untuk membaca tabel *look-up* tabel program.

#### 2.6.3.6.4. Konstanta *Immediate*

Pengalamatan langsung dilakukan dengan memberikan nilai ke suatu register secara langsung dengan menggunakan tanda #.

Contoh : `mov a,#100h`

#### 2.6.3.7. Mekanisme Pemasukkan Program ke AT 89C2051

AT89C2051 dikirimkan dengan 2K byte pada chip PEROM, kode memori *array* dalam status terhapus, (yaitu= FFH) dan siap untuk diprogram. Kode memori *array* diprogramkan per-byte secara serentak. *Array* diprogramkan hanya sekali, untuk memprogram kembali byte yang terisi, keseluruhan *array* memori harus dihapus secara elektrik.

##### 2.6.3.7.1. *Internal Address Counter*

AT89C2051 mempunyai suatu PEROM internal yang selalu reset lagi ke 000H pada penaikan RST taitu dengan memberikan pulsa positif pada pin XTAL1.

##### 2.6.3.7.2. *Algoritma Programming*

Untuk memrogram AT89C2051 yang mempunyai suatu PEROM internal yang selalu reset lagi ke 000H pada penaikan RST dengan memberikan pulsa positif pada pin XTAL1 Urutannya adalah sebagai berikut :

##### 1. Urutan *Power-Up*:

Menerapkan power antara pin VCC dan GND

RST dan XTAL1 di-set ke GND

##### 2. Pin RST di-set ke H.

Pin P3.2 di-set ke H.

3. Menyesuaikan kombinasi H ke L logic dari pin P3.3, P3.4, P3.5, P3.7 untuk memilih salah satu operasi pemrograman pada PEROM *programming Mode Table to program and verify the array*.
  4. Menyiapkan data untuk kode byte pada lokasi 000H pada P1.0 sampai P1.7 .
  5. Menaikkan RST ke 12V untuk *enable* program.
  6. Pulsa P3.2 sekali ketika memprogram suatu byte pada PEROM *array* atau untuk *lock* bit. Waktu *cycle byte write* adalah diatur 1,2 m s.
  7. Untuk memverifikasi data yang diprogramkan, menurunkan RST dari 12V ke logika II dan mengeset pin P3.3 sampai 3.7 ke level yang sesuai. Data output dapat dibaca dpada port pin P1.
  8. Untuk meprogram byte pad lokasi alamat yang berikutnya, dengan memberikan pulsa pada pin XTAL1 yang merupakan *internal address counter*. Menyiapkan data baru pada port pin P1.
  9. Mengulangi langkah-langkah 6 sampai 8, mengubah data dan menempatkan address counter untuk keseluruhan 2K byte array atau sampai akhir seluruh *file* dicapai.
  10. Urutan Power-Off:
    - XTAL1 di-set ke L
    - RST di-set ke L
    - Matikan Vcc
-

#### 2.6.3.7.3. Program verifikasi

1. Mereset *internal address counter* pada 000H oleh RST dari L ke H
2. Menyesuaikan kontrol sinyal untuk membaca kode data dan membaca output data pada port P1.
3. Menberikan pulsa pada pin XTAL1 sekali untuk menentukan alamat *internal address counter*.
4. Membaca kode data berikutnya yang berikutnya pada port P1
5. Mengulangi langkah-langkah 3 dan 4 sampai keseluruhan *array* terbaca.

*Lock* bit tidak bisa diverifikasi secara langsung. Verifikasi dari *lock* bit dicapai dengan pengamatan bahwa kedudukannya *enable*.

#### 2.6.3.7.4. Chip Erase





Keseluruhan PEROM *array* (2K byte) kedua *lock* bit dihapus secara elektrik dengan menggunakan kombinasi sinyal kontrol yang sesuai dan dengan pemilikan P3.2 yang rendah pada 10 ms. Kode *array* ditulis dengan semua 1 s di dalam chip pada operasi penghapusan dan harus dieksekusi sebelum byte memori yang terisi byte diprogram kembali.

#### 2.6.3.7.5. Reading The Signature Byte

*Signature hYTE* akan membaca sampai prosedur pada verifikasi normal pada lokasi 000H, 001H, dan 002H, kecuali pada P3.5 dan P3.7 harus di-*pull* ke logika rendah.

(000H) = 1E Menunjukkan dibuat oleh Atmel

(001H) = 21H menunjukkan 89C2051

Mode		RST/V <sub>pp</sub>	P3.2/ $\overline{\text{PROG}}$	P3.3	P3.4	P3.5	P3.7
Write Code Data <sup>16-18</sup>		12V		L	H	H	H
Read Code Data <sup>19</sup>		H	H	L	L	H	H
Write Lock	Bit-1	12V		H	H	H	H
	Bit-2	12V		H	H	L	L
Chip Erase		12V		H	L	L	L
Read Signature Byte		H	H	L	L	L	L

Gambar 2.12

Mode *Flash Programming*<sup>131</sup>

2.7. Modul FT232 USB Serial Converter

Perkembangan teknologi komputer dan antarmuka (*interface*) didasari atas kebutuhan dan perkembangan ilmu pengetahuan. Teknologi antarmuka pada komputer kini mulai beralih dari port serial RS232 ke USB dan antarmuka lain yang menyediakan kecepatan pengiriman data jauh lebih cepat seperti bluetooth, WiFi dan lain-lain. Komputer pada umumnya memiliki lebih dari dua port USB. Hal ini disebabkan banyaknya devais dan aksesoris komputer yang menggunakan antarmuka USB sebagai komunikasinya. Adapun contoh pengiriman informasi secara serial melalui sebuah mikrokontroler yang dikirimkan ke PC melalui port USB seperti Modul FT232-USB yaitu modul *interface* yang digunakan untuk aplikasi dari mikrokontroller ke USB, komunikasi ini dilakukan secara serial dengan kata lain sebuah modul yang dapat mengkonversi data serial yang berasal



dari mikrokontroller. Secara umum modul ini berfungsi untuk mengubah data USB yang berasal dari port USB menjadi data serial dengan level tegangan TTL sehingga pengguna dapat melakukan komunikasi data serial (UART) melalui port USB. Keunggulan digunakannya modul ini adalah kemampuannya untuk mengirim data lebih cepat daripada komunikasi serial dengan menggunakan Port RS232 dan dimensinya yang ringkas. Untuk kecepatan transfer data serial yang dapat di pakai oleh IC ini yaitu sebesar 300bps sampai 9600 bps.

Modul FT232 Serial-USB *Converter* mempunyai spesifikasi sebagai berikut:

1. Memiliki tegangan kerja 4,4- 5,25 volt DC.
2. Tersedia 2 LED untuk indikator Tx dan Rx data pada komunikasi serial.
3. Memiliki baudrate 3Mbps (TTL), 1Mbps (RS-232), 3Mbps (RS-422/RS-485).
4. Pin sinyal kontrol (arah) untuk komunikasi RS-485 yang bekerja secara otomatis.
5. Kompatibel dengan USB 1.1 dan USB 2.0.

Modul menggunakan konfigurasi daya *Self Powered*.

6. Memiliki *output* dengan level TTL 5 volt .
  7. Memiliki EEPROM eksternal untuk menyimpan data PID, VID, nomor serial dan deskripsi produk pengisian datanya melalui USB.
  8. Virtual COM port driver (VCP) dan D2xx (USB *Direct Drivers* + D.L. *S/W Interface*) untuk Windows 98, 98SE, ME, 2000 dan XP.
  9. Mendukung format UART dengan 7/8 bit data , 1 / 2 stop bit dan *Odd/Even /Mark/Space/No Parity*.
-

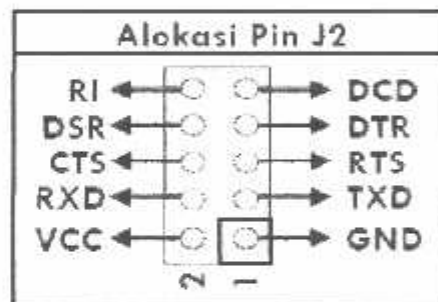
Adapun gambar fisik dari modul FT232 Serial-USB *Converter* sebagai berikut:



Gambar 2.13

### Modul FT232 Serial-USB *Converter* <sup>[1]</sup>

Gambar diatas merupakan modul dari FT232-USB secara fisik yang terdiri dari konektor USB TIPE B dan chip FT232 sebagai komponen utama dari modul COM to USB tersebut serta komponen-komponen pendukung lainnya.

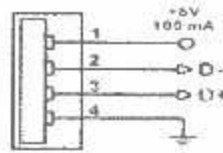


Gambar 2.14

Alokasi pin yang digunakan pada modul FT232 USB Serial *Converter* <sup>[1]</sup>

#### 2.7.1. Kabel USB

Untuk menghubungkan modul FT232 Serial-USB *Converter* dengan komputer maka diperlukan kabel konektor USB dimana hanya terdapat ada 2 macam konektor yang digunakan dalam menghubungkan modul tersebut, yaitu konektor type A dan konektor type B seperti terlihat dalam Gambar 1. Konektor



**Gambar 2.16**

### **Konektor Pin USB Tipe A <sup>[2]</sup>**

#### **2.7.2. Sinyal USB**

Secara umum USB terdiri dari 4 buah pin yaitu pada pin 1 merupakan sumber tegangan 5 Volt , pin nomor 2 merupakan pin D+ serta pin nomor 3 merupakan pin D- dimana pin nomor 2 dan pin nomor 3 merupakan jalur pengiriman dan penerima data yang berasal dari mikrokontroller sedangkan pin nomor 4 merupakan ground.

Sinyal digital yang dikirim melalui dua saluran tersebut dikatakan sebagai '*difference signal*', artinya sinyal digital '0' atau '1' tidak dinyatakan dengan besarnya tegangan pada saluran tersebut terhadap ground, seperti halnya sinyal digital yang dipakai dalam IC TTL (*Transistor Transistor Logic*) atau dalam saluran RS232.

Sinyal digital dinyatakan dengan perbedaan tegangan antara dua kabel tersebut. Jika tegangan pada saluran D+ lebih tinggi dari tegangan pada saluran D- maka informasi yang dikirimkan adalah sinyal digital '1', sebaliknya sinyal digital '0' dinyatakan dengan tegangan pada D+ lebih kecil dari D-.

type A dipakai untuk menghubungkan kabel USB ke terminal USB yang ada pada bagian komputer sedangkan konektor tipe B dipakai untuk menghubungkan kabel USB ke modul FT232 Serial-USB *Converter*. Sedangkan untuk peralatan USB yang sederhana, misalnya mouse, biasanya tidak memakai konektor B melainkan konektor tipe A hal ini dilakukan untuk menghemat biaya sehingga kabel langsung dihubungkan ke bagian dalam mouse.



**Gambar 2.15**

### **Konektor USB<sup>[2]</sup>**

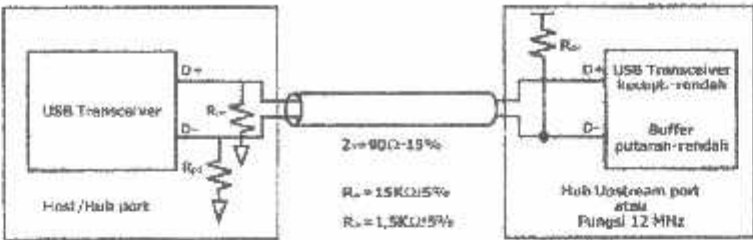
Kabel USB terdiri dari 4 kabel ditambah konduktor pembungkus kabel. Kabel nomor 1 dipakai untuk menyalurkan sumber daya dengan tegangan 5 Volt. jika diperlukan peralatan USB boleh mengambil daya dari saluran ini tidak lebih dari 100 mA. Komputer yang dilengkapi dengan antarmuka USB, wajib menyediakan daya sebesar 500 mA untuk keperluan ini. Peralatan USB yang memerlukan daya lebih dari ketentuan tersebut di atas, harus menyediakan sendiri sumber daya untuk keperluan kerja peralatan tersebut.

Kabel nomor 4 adalah ground sebagai saluran balik sumber tegangan 5 Volt. Kabel nomor 2 dan nomor 3 dipakai untuk pengiriman sinyal. Kabel nomor 2 bernama D<sup>-</sup> dan kabel nomor 3 bernama D<sup>+</sup>, tegangan pada dua saluran ini berubah antara 0 Volt dan 3,3 Volt seperti terlihat pada gambar 2 berikut:

Rentang tegangan USB adalah 0,3 volt hingga 3,6 volt (pada beban 1,5 k $\Omega$ ) logika tinggi didapat jika tegangan sudah melebihi 2,8 volt terhadap ground pada beban 1,5 k $\Omega$ . Pada piranti USB yang berkecepatan rendah dan penuh untuk deferensial “1” dikirim dengan menarik D+ hingga lebih besar dari 2,8 volt dengan sebuah resistor 1,5 k $\Omega$  terhubung ke ground dan sekaligus menarik D- hingga dibawah 0,3 volt dengan dengan sebuah resistor 1,5 k $\Omega$  terhubung ke 3,6 volt lebih rendah dari 0,3 volt dengan resistor pull-up dan pull-down yang sama. Di bagian penerima, deferensial “1” dideferensialkan sebagai D+ lebih besar 200mV dari D-, dan diferensial “0” berarti D+ lebih kecil dari 200mV dibanding D-. Pada USB berkecepatan tinggi (480MBit/s) digunakan sumber arus tetap mA untuk mengurangi *noise*.

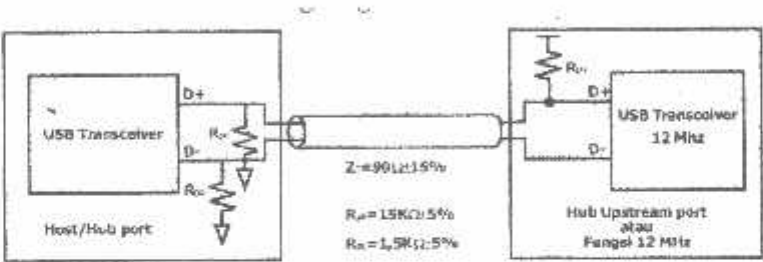
Kecepatan piranti USB dibagi menjadi dua yaitu kecepatan rendah (low-speed) dan kecepatan tinggi (full-speed) yang dihubungkan dengan cara koneksi resistor dan kabel USB oleh posisi resistor *pull-up* di ujung kabel *downstream* yaitu sebagai berikut :

- Piranti *full-speed* diterminalkan dengan resistor pull-up terhubung di D+ seperti gambar 2.17a.
  - Piranti low-speed diterminalkan dengan resistor pull-up terhubung di D-, seperti gambar 2.17b.
  - *Terminator pull-down* di port *down-stream* adalah resistor 15k $\Omega$   $\pm$ 5% terhubung ke *ground*.
-



Gambar 2.17a.

Perkabelan USB Kecepatan Rendah<sup>[2]</sup>



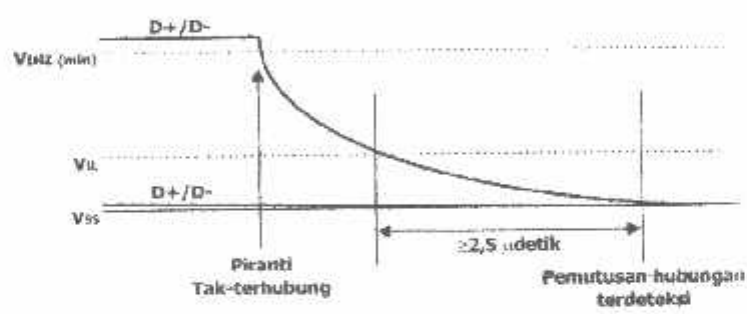
Gambar 2.17b.

Perkabelan USB kecepatan penuh<sup>[2]</sup>

2.7.3. Sinyal Penyambungan dan Pemutusan

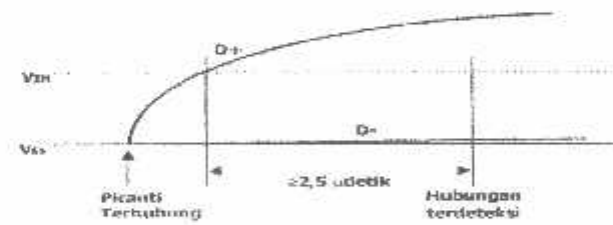
Pada sinyal penyambungan dan pemutusan USB yaitu ketika tidak ada piranti terhubung ke host atau komputer maka pull-down resistor yang ada akan menyebabkan D+ dan D- tertarik hingga di bawah ambang logika rendah host atau komputer ,hal ini menyebabkan munculnya keadaan SE0 (*Single-Ended 0*) di port downsteram. Kondisi tak terhubungnya suatu piranti USB dari port akan dideteksi jika *host* atau komputer tidak *men-drive header* (di jalur data) selama lebih dari 2,5  $\mu$ detik.

Sedangkan jika sebuah piranti dihubungkan, maka host atau komputer akan mendeteksi bahwa salah satu jalur datanya tertarik hingga lebih besar dari ambang  $V_{IH}$  selama lebih dari 2,5  $\mu$ detik. *Host* kemudian bisa (*option*) mendeteksi kecepatan piranti yang baru terhubung dengan men-sempel keadaan bus segera sebelum men-*drive* SE0 untuk me-*reset* piranti. Jika diinginkan, host dapat membuat bus mengambang sesudah meresetnya dan menjalankan evaluasi bus sesudah 2,5  $\mu$ detik tersebut. Lebih jelasnya seperti tampak pada gambar berikut :



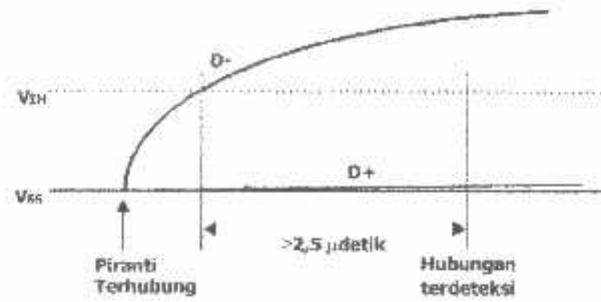
Gambar 2.18

**Sebuah piranti dilepas dari port USB<sup>[2]</sup>**



Gambar 2.19

**Sebuah piranti USB kecepatan tinggi dihubungkan ke port USB pada  
*host/komputer* <sup>[2]</sup>**



Gambar 2.20

**Sebuah piranti USB kecepatan rendah dihubungkan ke port USB pada  
host/komputer<sup>[2]</sup>**

Untuk mengirimkan paket data, USB menerapkan encode data NRZI (*Non Return to Zero Invert*). Dalam NRZI ini logika "1" berarti tidak ada perubahan level tegangan dan logika "0" ditunjukkan dengan adanya perubahan level tegangan. Paket data dikirimkan ke bus USB berurutan dari bit yang berbobot paling rendah (LSB, *Least Significant Bit*), diikuti LSB berikutnya dan terakhir adalah bit yang berbobot paling tinggi (MSB, *Most Significant Bit*).





### BAB III

#### PERANCANGAN DAN PEMBUATAN ALAT

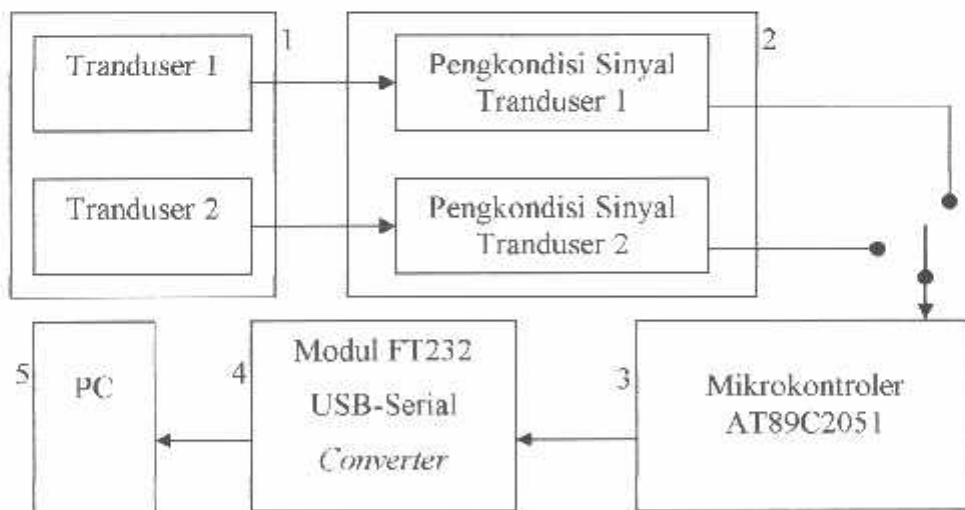
Pada dasarnya perencanaan dan pembuatan alat ini secara garis besar dibagi atas dua bagian, yaitu:

1. Perancangan perangkat keras
2. Perancangan perangkat lunak

Komponen yang dipakai dalam perencanaan ini antara lain mikrokontroler AT89C2051 sebagai kontrol utama, dengan komponen pendukung meliputi *microphone*, rangkaian pengkondisi sinyal dan modul FT232 USB-Serial *Converter* sebagai antarmuka dengan PC.

#### 3.1. Pendahuluan

Untuk mendeskripsikan keseluruhan sistem dapat dilihat dari diagram blok di bawah ini.



Gambar 3.1

Diagram Blok Sistem

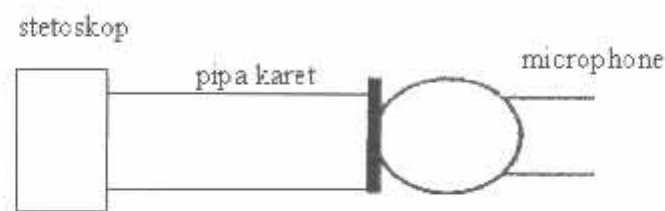
Keterangan tiap blok:

1. Transduser ; transduser pertama yang digunakan adalah *microphone* dengan bantuan stetoskop, dimana transduser ini berfungsi mendeteksi suara denyut jantung. Transduser kedua menggunakan fotodioda dengan bantuan cahaya LED (*Light Emitting Diode*) yang mendeteksi denyut jantung dari aliran darah pada jari tangan.
  2. Pengkondisi Sinyal ; rangkaian pengkondisi sinyal yang digunakan untuk transduser *microphone* dan stetoskop adalah rangkaian penyangga (*buffer*), rangkaian penguat tak membalik (*non-inverting amplifier*) dan rangkaian pembanding tegangan (*voltage comparator*), sedangkan rangkaian pengkondisi sinyal untuk transduser fotodioda dengan LED menggunakan sebuah rangkaian pengubah arus ke tegangan (*current to voltage converter*), rangkaian penyangga (*buffer*) dan rangkaian pembanding tegangan (*voltage comparator*). Fungsi rangkaian ini adalah untuk menyesuaikan tegangan keluaran transduser dengan level tegangan mikrokontroler.
  3. Mikrokontroler AT89C2051 ; digunakan sebagai kendali utama sistem untuk mengolah data yang diterima, mengontrol dan mengirimkan hasilnya ke *Personal Computer* (PC).
  4. Modul FT232 USB-Serial *Converter* ; sebuah modul berbasis chip FT232 yang berfungsi mengkonversi pengiriman data secara serial biasa menjadi serial USB sehingga mikrokontroler dapat dikomunikasikan dengan PC secara serial melalui antarmuka USB
-

5. PC ; digunakan untuk menampilkan hasil penghitungan denyut jantung. Berisi pilihan usia orang yang akan diukur. Menampilkan hasil penghitungan serta kondisi dari hasil penghitungan apakah hasilnya normal atau tidak normal.

### 3.2. Perancangan Perangkat Keras

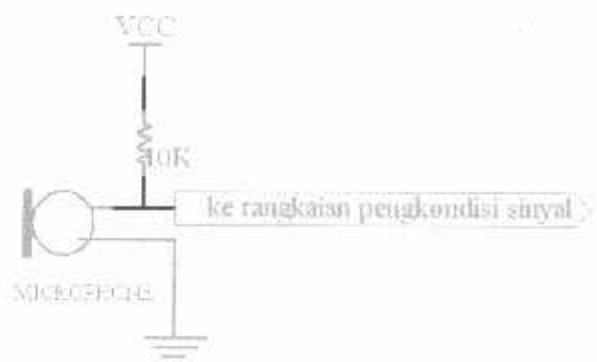
#### 3.2.1 Perancangan Tranduser



**Gambar 3.2**

#### *Tranduser Akustik*

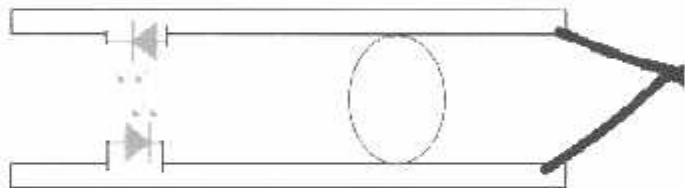
Tranduser pertama yang digunakan adalah sebuah *microphone* yang disambung dengan sebuah stetoskop akustik. Bagian utama stetoskop yang paling berperan adalah diafragma yang menerima langsung suara denyut jantung. Suara dari diafragma akan diteruskan ke *microphone* melalui pipa karet yang menghubungkan diafragma stetoskop dengan *microphone*. Suara yang diterima oleh *microphone* akan diubah menjadi sinyal tegangan. Sinyal tegangan ini yang akan dikuatkan oleh rangkaian pengkondisi sinyal.



Gambar 3.3

Rangkaian *Microphone*

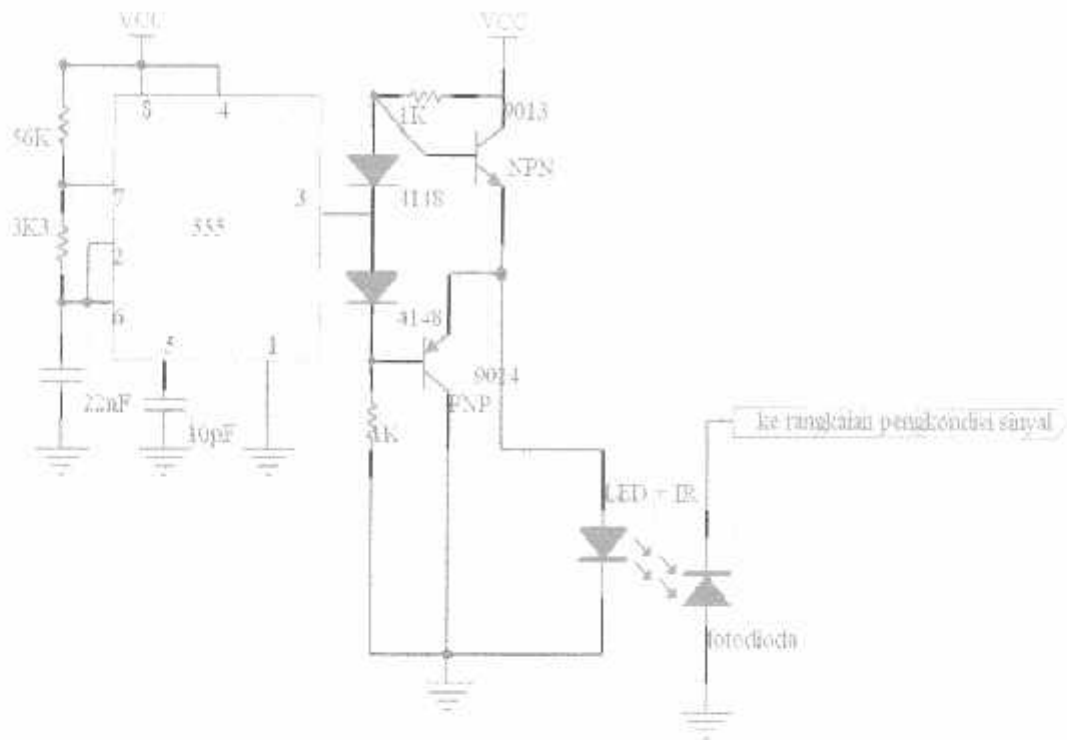
Saat suara diterima oleh *microphone* maka akan mengubah tegangan yang masuk ke pengkondisi sinyal sehingga perubahan suara akan mengakibatkan perubahan tegangan yang akan mengindikasikan denyut yang diterima oleh tranduser.



Gambar 3.4

Plesthymograf

Tranduser kedua yang digunakan adalah sebuah fotodioda yang diposisikan berhadapan dengan LED dalam penjepit untuk menahan posisi jari. Keluaran fotodioda berupa arus yang berubah-ubah sesuai intensitas cahaya yang diterimanya. Arus inilah yang akan diubah menjadi tegangan.



Gambar 3.5

### Rangkaian LED dan Fotodiode

Karena transduser plesthymograf memakai satu LED yang berisi dua sumber spectrum dengan polaritas saling terbalik. Anoda LED inframerah bersatu dengan katoda LED merah sedangkan katoda LED inframerah bersatu dengan LED inframerah sehingga diperlukan rangkaian balans pengikut *emitter* yang diberi pulsa oleh *timer* 555. Sehingga nyala kedua LED bergantian sangat cepat.

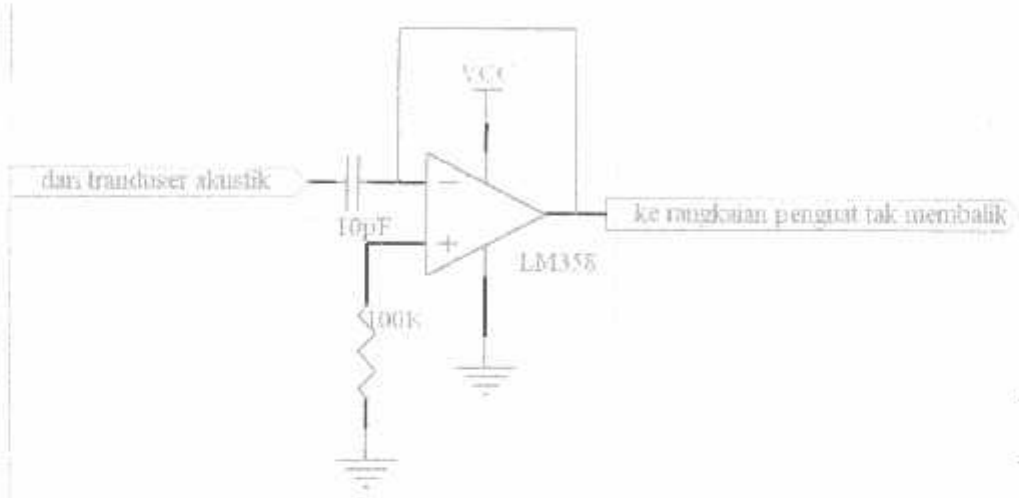
### 3.2.2. Perancangan Rangkaian Pengkondisi Sinyal

#### 3.2.2.1. Rangkaian Pengkondisi Sinyal Transduser Akustik

##### 3.2.2.1.1. Rangkaian Penyangga (*Buffer*)

Rangkaian pertama merupakan rangkaian *buffer* atau penyangga. Berfungsi untuk menyangga agar sinyal yang lewat tidak

melemah. Rangkaian ini menggunakan LM358. Detak jantung yang diterima tranduser akustik yang kemudian diubah menjadi sinyal listrik oleh *microphone* menghasilkan sinyal tegangan maksimal mencapai 2 mV.



Gambar 3.6

Rangkaian Penyangga (*Buffer*)

3.2.2.1.2 Rangkaian Penguat Tak Membalik (*Non-Inverting Amplifier*)

Rangkaian kedua berupa rangkaian penguat tak membalik atau *non-inverting amplifier*. Masukan penguat ini masuk ke pin 3 (+) kemudian dikuatkan lebih dari dua kali. Rangkaian penguat tak membalik ini menggunakan LM358. Tegangan keluaran yang diinginkan adalah 2 volt. Untuk mencapai tegangan yang diinginkan maka diperlukan penguatan sebanyak 1000 kali.

$$Vo = \frac{Av}{Vi}$$

$$2 = \frac{Av}{0.002}$$

$$A_v = 1000 \text{ kali}$$

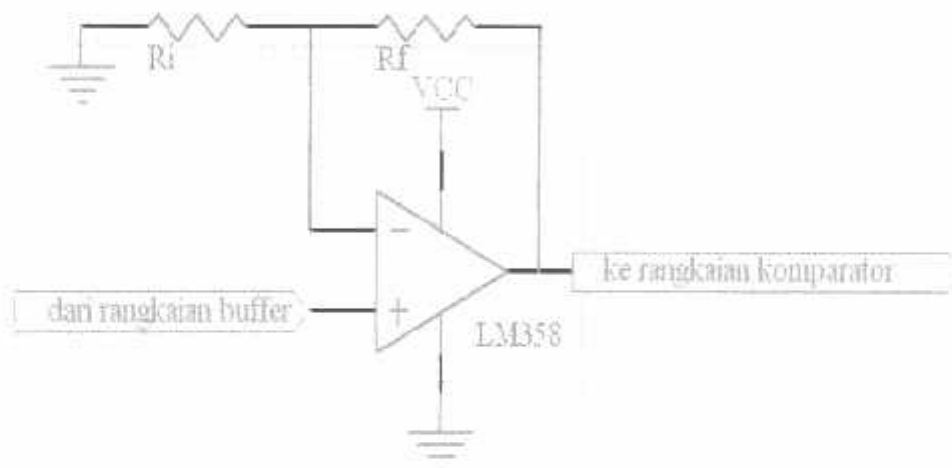
$$A_v = \frac{R_f}{R_i} + 1$$

$$1000 = \frac{R_f}{R_i} + 1$$

Jika  $R_i = 1 \text{ K}\Omega$ , maka

$$R_f = (1000 \times 1 \text{ K}\Omega) + 1$$

$$R_f \approx 1 \text{ M}\Omega$$



Gambar 3.7

### Rangkaian Penguat Tak Membalik (*Non-Inverting Amplifier*)

#### 3.2.2.1.2. Rangkaian Pembanding Tegangan (*Voltage Comparator*)

Rangkaian pembanding tegangan atau komparator menggunakan LM311. Untuk tegangan referensi digunakan 5 V dan nilainya disesuaikan dengan amplitudo tegangan masukan ke komparator diatur dengan menggeser nilai tahanan variabel 50 k $\Omega$ . Pada percobaan  $V_{ref}$  diset pada 2 Volt dikarenakan tegangan masukan denyut jantung dari penguat tak membalik sebesar 2.1 Volt



Bila tegangan  $V_{in}$  bergeser lebih dari  $V_{ref}$  maka keluaran menuju  $V_{sat+}$  dan jika sebaliknya maka keluaran menuju  $V_{sat-}$ .

$$V_{out} = V_{ref} - V_{in}$$

$$V_{out} = V_{sat+}, \text{ jika } V_{in} > V_{ref} \text{ dan}$$

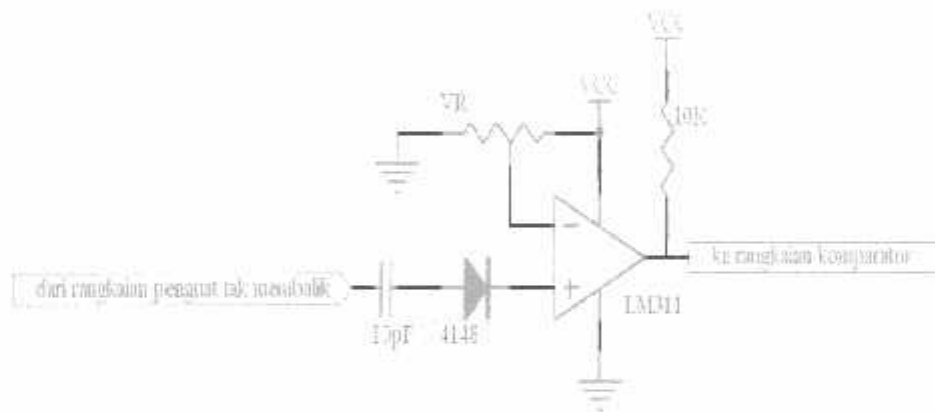
$$V_{out} = V_{sat-}, \text{ jika } V_{in} < V_{ref}$$

Jika tegangan masukan sebesar 2.1 V dan tegangan referensi sebesar 2 V maka;

$$V_{out} = 2 \text{ V} - 2.1 \text{ V}$$

$$V_{out} = V_{sat-} = -0.1 \text{ V}$$

Dengan adanya tegangan masukan 2.1 V maka akan menghasilkan keluaran '1'.



**Gambar 3.8**

### **Rangkaian Pembanding Tegangan**

#### **3.2.2.1. Rangkaian Pengkondisi Sinyal untuk Transduser Plesthymograf**

##### **3.2.2.2.1. Rangkaian Pengubah Arus ke Tegangan (*Current to Voltage Converter*)**

Rangkaian pertama berupa rangkaian penyangga atau *buffer*. Rangkaian ini menggunakan IC LF351. Karena sensitivitas fotodiode

merupakan besaran arus (A/W) maka tegangan keluaran didapatkan dari rumus  $V_o = -I R_L$ . Fotodioda biasanya dioperasikan *bias reverse*. Dengan perubahan tegangan sebesar 0.5 – 1.5 Volt maka arus keluaran fotodioda didapat sekitar -0.22 sampai -0.07  $\mu\text{A}$  maka diperlukan resistansi beban sebesar 7 M $\Omega$ . Resistansi yang digunakan adalah sebuah resistor 4.7 M $\Omega$  dan sebuah resistor 2.2 M $\Omega$  yang disusun secara seri hingga resistensi totalnya mendekati 7 M $\Omega$  (6.9 M $\Omega$ ).

$$V_{o1} = -I R_L$$

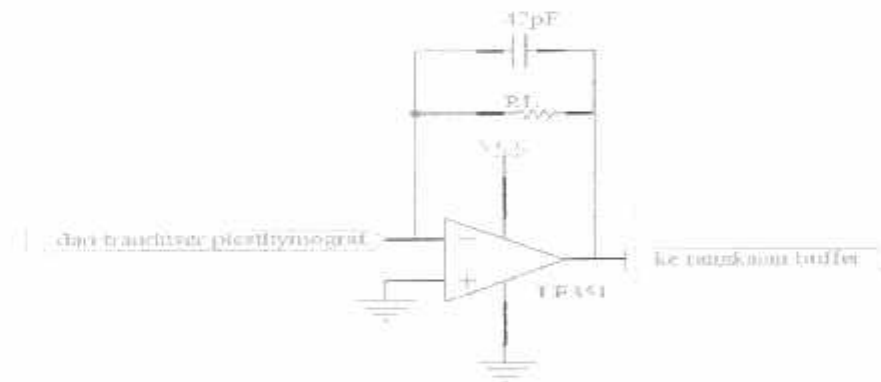
$$V_{o1} = -(-0.22 \times 10^{-6}) \times 6.9 \times 10^6$$

$$V_{o1} \approx 1.5 \text{ Volt}$$

$$V_{o2} = -I R_L$$

$$V_{o2} = -(-0.07 \times 10^{-6}) \times 6.9 \times 10^6$$

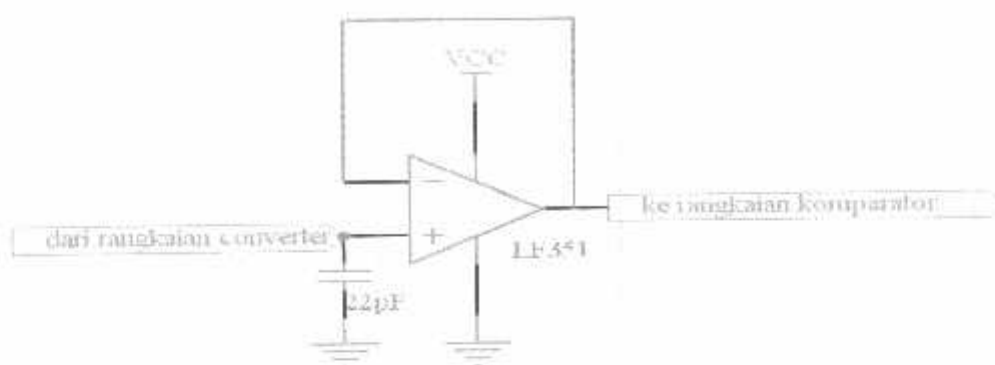
$$V_{o2} \approx 0.5 \text{ Volt}$$



Gambar 3.9

#### Rangkaian Pengubah Arus ke Tegangan (*Current to Voltage Converter*)

### 3.2.2.2.2. Rangkaian Penyangga (*Buffer*)



Gambar 3.10

### Rangkaian Penyangga (*Buffer*)

### 3.2.2.2.3. Rangkaian Pembanding Tegangan (*Voltage Comparator*)

Rangkaian pembanding tegangan atau komparator menggunakan LM311. Untuk tegangan referensi digunakan 5 V dan nilainya disesuaikan dengan amplitudo tegangan masukan ke komparator diatur dengan menggeser nilai tahanan variabel 10 k $\Omega$ . Pada percobaan  $V_{ref}$  diset pada 1 Volt dikarenakan tegangan masukan denyut jantung dari *buffer* sebesar 1.5 Volt

Bila tegangan  $V_{in}$  bergeser lebih dari  $V_{ref}$  maka keluaran menuju  $V_{sat+}$  dan jika sebaliknya maka keluaran menuju  $V_{sat-}$ .

$$V_{out} = V_{ref} - V_{in}$$

$$V_{out} = V_{sat+}, \text{ jika } V_{in} > V_{ref} \text{ dan}$$

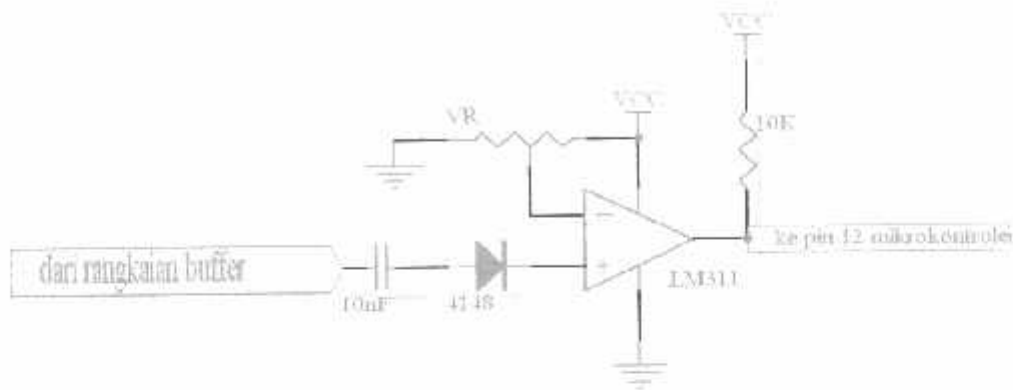
$$V_{out} = V_{sat-}, \text{ jika } V_{in} < V_{ref}$$

Jika tegangan masukan sebesar 1.5 V dan tegangan referensi sebesar 1 V maka;

$$V_{out} = 2 \text{ V} - 1.5 \text{ V}$$

$$V_{out} = V_{sat-} = -0.5 \text{ V}$$

Dengan adanya tegangan masukan 1,5 V maka akan menghasilkan keluaran '0'.



Gambar 3.11

### Rangkaian Pembanding Tegangan

#### 3.2.3. Mikrokontroller AT89C2051

Mikrokontroler AT89C2051 memiliki 20 kaki (pin). Berikut ini pin-pin yang digunakan beserta rangkaian *oscillator* dan rangkaian *reset*. Rangkaian *oscillator* berfungsi untuk membangkitkan siklus mesin sedangkan rangkaian *reset* berfungsi untuk mengarahkan program ke alamat 00H

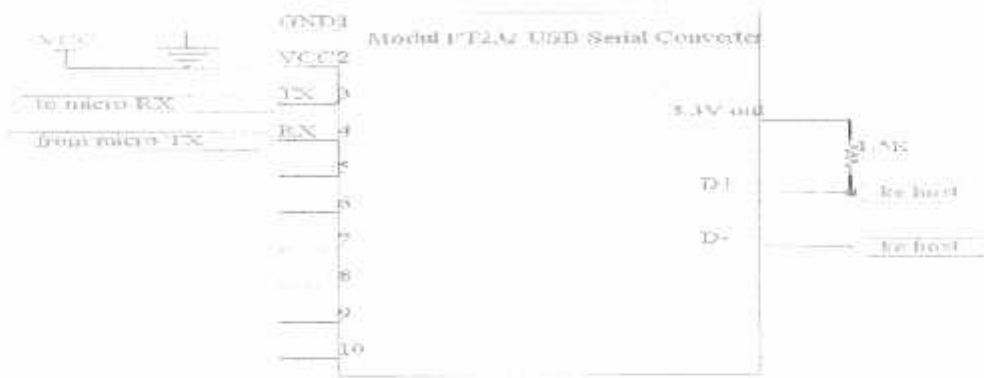


Gambar 3.12

### Alokasi pin mikrokontroler AT89C2051

### 3.2.4. Modul FT232 USB-Serial Converter

Modul FT232 USB-Serial Converter digunakan untuk berkomunikasi secara serial dengan mikrokontroler dengan menghubungkan pin RX ke pin 3 (TX) pada mikrokontroler dan pin TX ke pin 2 (RX) pada mikrokontroler. Juga memberi catu daya DC sebesar 5 V pada pin 1 dan GND pada pin 2.

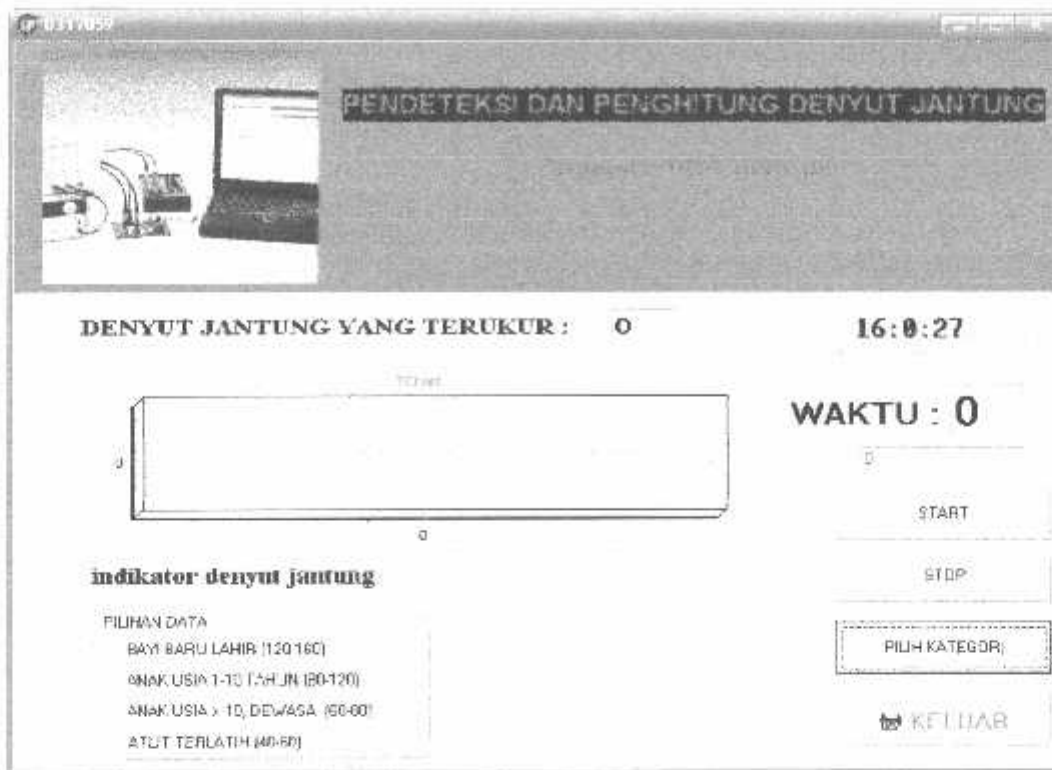


Gambar 3.13

### Rangkaian pin modul FT232 USB Serial Converter

### 3.2.5. Personal Computer (PC)

PC digunakan untuk menampilkan hasil pengukuran serta mengatur tombol-tombol pilihan untuk proses penghitungan denyut jantung. Pemrograman pada PC menggunakan bahasa pemrograman Delphi.



Gambar 3.14

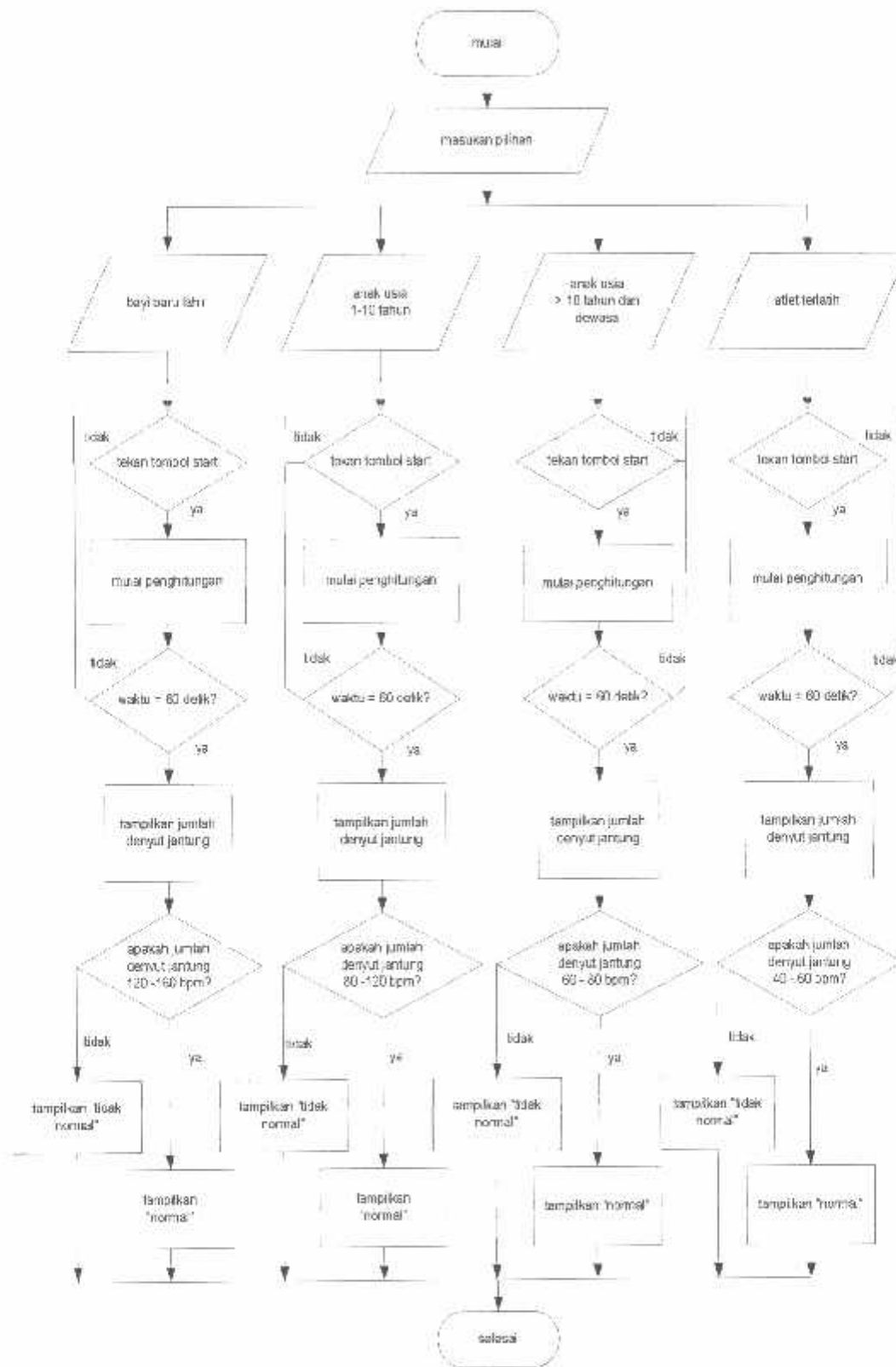
### Perencanaan Tampilan pada PC

Orang yang akan diukur memilih kategori, kemudian setelah transduser terpasang tekan tombol *start*. Saat tombol *start* ditekan waktu akan mulai menghitung sampai 60 detik, kemudian akan muncul hasil denyut jantung yang terukur. Hasil denyut jantung yang terukur akan mengindikasikan apakah denyut jantung yang terukur normal atau tidak normal sesuai kategori yang dipilih.

### 3.3. Perancangan Perangkat Lunak

Perencanaan perangkat lunak akan menampilkan diagram alir (*flow chart*) yang akan menggambarkan cara kerja sistem.

### 3.3.1. Flow Chart Keseluruhan Sistem

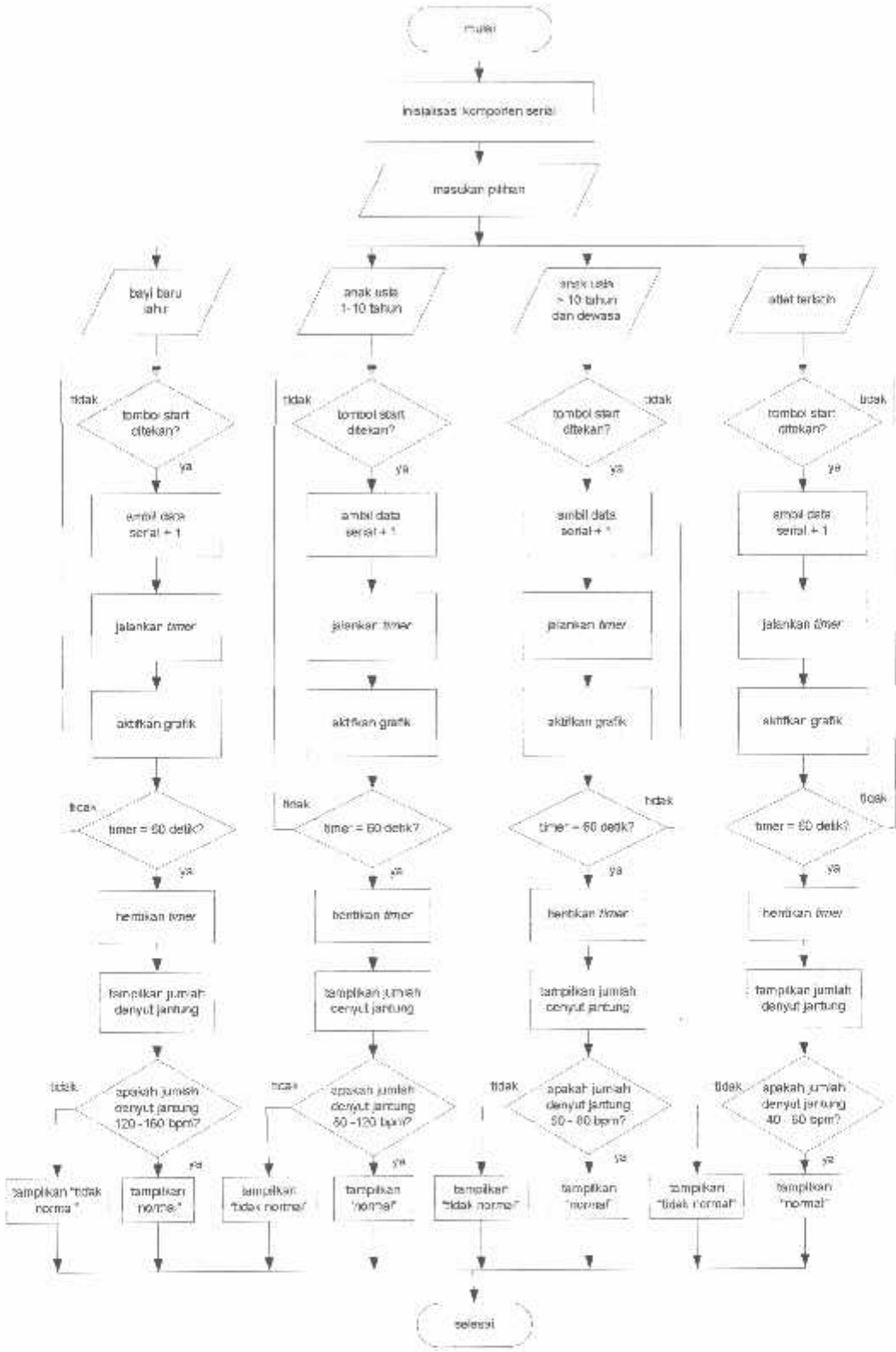


3.3.2 Flow Chart Mikrokontroler





3.3.2 Flow Chart di Personal Computer (PC)



## **BAB IV**

### **PENGUJIAN ALAT**

#### **4.1. Pendahuluan**

Dalam bab ini membahas tentang pengujian dari modul yang digunakan. Secara umum pengujian ini bertujuan untuk mengetahui apakah piranti yang telah direalisasikan dapat bekerja sesuai dengan perencanaan yang telah direncanakan.

#### **4.2. Tujuan Pengujian**

Tujuan pengujian pada masing-masing blok pada perancangan dan pembuatan pendeteksi dan penghitung denyut jantung manusia ini bertujuan:

- Untuk mengetahui apakah tiap blok rangkaian dapat berfungsi dengan baik
- Untuk mengetahui sistem kerja pendeteksi dan penghitung denyut jantung manusia dengan menggunakan mikrokontroller AT89C2051 sebagai pengendali utama.

#### **4.3. Pengujian Modul FT232 USB Serial Converter**

Tujuan dari pengujian ini adalah untuk mengetahui apakah modul USB bisa digunakan untuk mengirim dan menerima data atau tidak. Adapun cara pengujianya dilakukan dengan cara memilih *COM* yang dideteksi oleh PC kemudian mencoba melakukan pengiriman data melalui program sederhana yang telah dibuat, data akan dikirim melalui kabel USB menuju Modul USB dengan menggunakan indikator apabila data di kirim dari PC ke USB maka led indikator

pada modul (Rx) akan menyala dan sebaliknya pada saat data dikirim kembali dari USB ke PC maka indikator pada modul (Tx) akan menyala, dengan prosedur pengujian seperti dibawah ini.



**Gambar 4.1**

### **Blok Pengujian Modul USB**

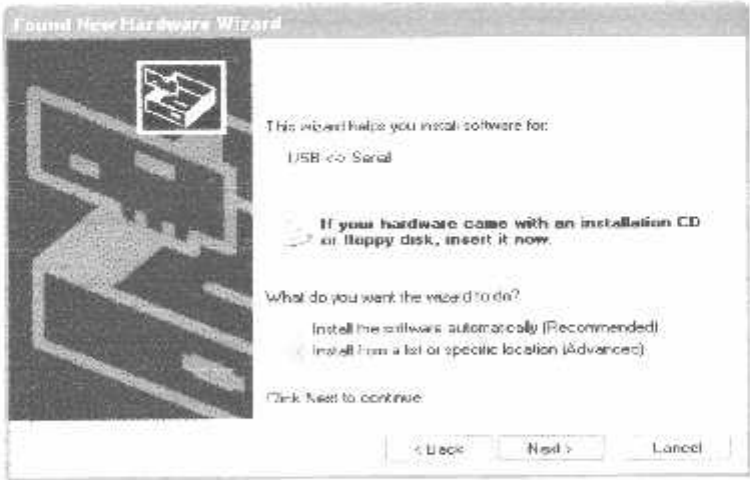
- Menghubungkan rangkaian Modul USB ke PC menggunakan kabel USB.
  - Menginstall driver modul USB.
  - Membuka program Delphi
1. Untuk melakukan proses penginstallan driver USB maka kita menghubungkan modul dengan PC menggunakan kabel konektor USB tipe A terhubung ke PC dan konektor tipe B terhubung pada Modul USB :
  2. Sebelum melakukan percobaan, terlebih dahulu kita menginstal driver Modul USB tersebut yaitu
    - Pada saat kabel konektor USB dihubungkan pada PC maka PC akan mendeteksi adanya perangkat baru seperti pada gambar 4,2:



Gambar 4.2

PC Mendeteksi *Hardware* baru.

- Setelah terdeteksi maka proses selanjutnya menekan Button *Next*, setelah *button* ditekan maka akan muncul seperti gambar 4.3 Perintah tersebut merupakan pemberitahuan apakah penginstallan dilakukan secara otomatis atau manual, tetapi pada saat menginstall dipilih yang manual:



Gambar 4.3

Perintah Untuk Memilih Cara Peng-*install-an*

- Proses selanjutnya saat *button Next* ditekan maka PC akan menampilkan perintah untuk memberitahu dimana lokasi driver USB tersebut seperti gambar 4.4 dan gambar 4.5 disini driver berada pada (drive G → PCLink USBer → Win98\_2000\_XP → Driver VCP) .



Gambar 4.4

#### Perintah Memberitahu Lokasi *Driver Modul USB.*



Gamhar 4.5

#### Lokasi *Driver*

- Apabila proses diatas telah selesai maka selanjutnya adalah proses peng-*install*-an dengan menekan *button Next* yang selanjutnya PC akan memproses *driver* tersebut seperti gambar 4.6 dan setelah proses *transfer* selesai maka akan tampil bahwa *driver* baru telah selesai di *install* seperti gambar 4.7.



Gambar 4.6

**Proses Transfer Driver USB.**



Gambar 4.7

**Proses Peng-*install*-an Selesai**

3. Setelah Proses peng-*install*-an selesai maka untuk melakukan pengujian terhadap modul

USB, digunakan sebuah program Delphi sederhana yang dibuat agar dapat digunakan untuk menguji modul USB tersebut.

❖ Program Uji Komunikasi Modul USB menggunakan Delphi :

unit USerial;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,  
Forms,

Dialogs, VaClasses, VaComm, StdCtrls;

type

TForm1 = class(TForm)

    Memo1: TMemo;

    VaComm9: TVaComm;

    Edit1: TEdit;

    Label1: TLabel;

    Button1: TButton;

    Label2: TLabel;

    procedure FormCreate(Sender: TObject);

    procedure VaComm1RxChar(Sender: TObject; Count: Integer);

    procedure Button1Click(Sender: TObject);

private

    { Private declarations }

```

    Pesan : String;

public
    { Public declarations }

end;

var
    Form1: TForm1;

implementation
    {SR *.dfm}

    procedure TForm1.FormCreate(Sender: TObject);
    begin
        VaComm1.Open;
    end;

    procedure TForm1.VaComm9RxChar(Sender: TObject; Count: Integer);
    var
        I: Integer;
        Tmp: string;
    begin
        //baca data yang diterima oleh VaComm
        Tmp := VaComm9.ReadText;
        for I := 1 to Length(Tmp) do
            case Tmp[I] of
                #10:; //lewatkan
                #13: //Tunggu Enter
            begin

```

---



```

Memo1.Lines.Add(Pesan);;

Pesan := ''; //reset Pesan

end;

else //bukan #10 atau #13

    Pesan := Pesan + Tmp[I];

end;

end;

```

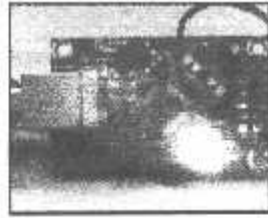
Program diatas sebagai percobaan mengirim data melalui PC ke modul USB di mana komponen yang digunakan pada delphi adalah

- *Vacomm* : komponen ini digunakan sebagai koneksi antara program Delphi dengan modul USB dan pada percobaan ini PC mendeteksi COM 6 untuk melakukan pengiriman
- *Edit* : Komponen ini digunakan sebagai input pada text memo.
- *Memo* : Komponen ini berfungsi untuk menampilkan data.
- *Button* : Komponen ini merupakan tombol untuk mengirim data

Sebelum program dijalankan terlebih dahulu *setting* COM yang dideteksi oleh PC setelah itu program di-*run* Tampilan pada saat program dijalankan dan nyala indikator TX dan RX pada modul FT232 USB Serial Converter saat data dikirim seperti pada.

---



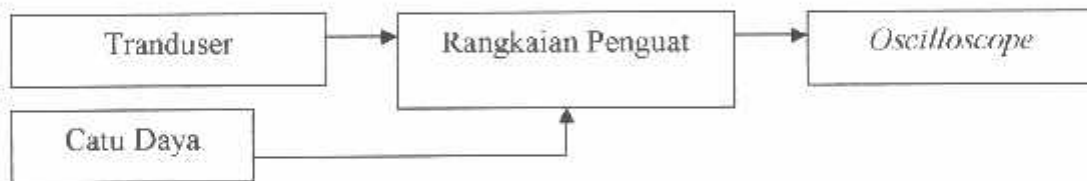


Gambar 4.10

Nyala Indikator pada Modul FT232 USB Serial Converter

#### 4.4. Pengujian Tranduser

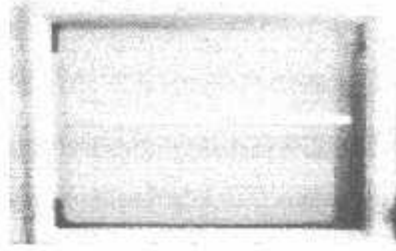
Tujuan dari pengujian ini adalah untuk mengetahui apakah seberapa besar masukan dari tranduser serta berapa besar penguatan yang diperlukan agar sinyal dari tranduser dapat diproses oleh mikrokontroler. Adapun prosedur pengujian tranduser adalah sebagai berikut.



Gambar 4.11

#### Blok Pengujian Tranduser

Untuk menampilkan sinyal dan perubahan sinyal tranduser diperlukan rangkaian penguat. Hal ini dikarenakan sinyal dari tranduser sangat kecil sehingga sulit untuk diamati.



Gambar 4.12

### Sinyal Keluaran Tranduser Akustik



Gambar 4.13

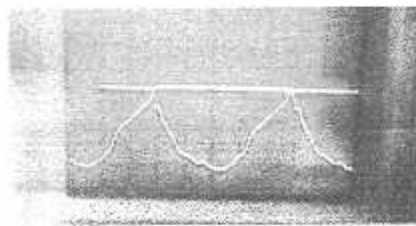
### Sinyal Keluaran Rangkaian Penguat Tranduser Akustik

Dengan penguatan 1000 kali didapatkan amplitudo mencapai 2 Volt. Sehingga sinyal yang diterima tranduser berkisar 2 mVolt.

$$A_v = \frac{V_{out}}{V_{in}}$$

$$1000 = \frac{2}{V_{in}}$$

$$V_{in} = 2 \text{ mVolt}$$



Gambar 4.14

### Sinyal Keluaran Tranduser Plesthymograf



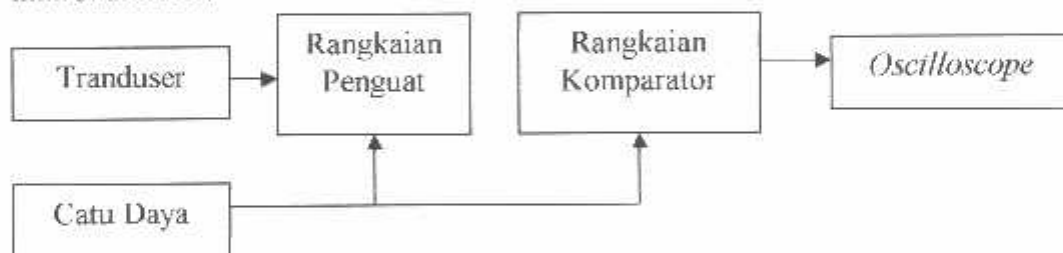
**Gambar 4.15**

#### **Sinyal Keluaran Rangkaian Penguat Transduser Plesthymograf**

Sinyal keluaran fotodioda tanpa penguatan menunjukkan bentuk sinyal dengan amplitudo mencapai 2 volt.

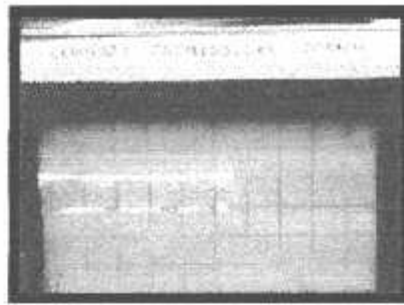
#### **4.5. Pengujian Rangkaian Pengkondisi Sinyal**

Tujuan dari pengujian ini adalah untuk mengetahui apakah rangkaian pengkondisi sinyal bekerja dengan baik menguatkan sinyal masukan dari transduser kemudian mengubahnya menjadi sinyal digital sebagai masukan mikrokontroler.



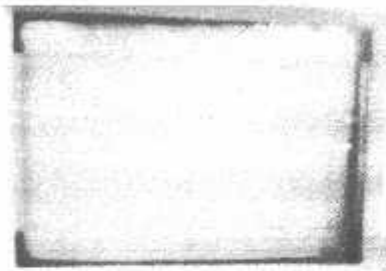
**Gambar 4.16**

#### **Blok Pengujian Rangkaian Pengkondisi Sinyal**



**Gambar 4.17**

**Sinyal Keluaran Rangkaian Komparator  
Tranduser Stetoskop dengan *Microphone***

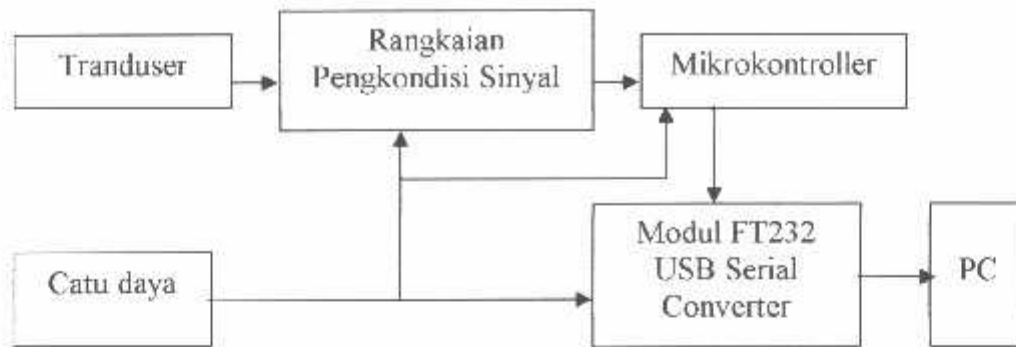


**Gambar 4.18**

**Sinyal Keluaran Rangkaian Komparator  
Tranduser Plesthymograf**

#### **4.6. Pengujian Keseluruhan Sistem**

Tujuan dari pengujian Sistem ini adalah untuk mengetahui apakah alat yang dibuat dan dirancang dapat bekerja sesuai yang diinginkan. Adapun prosedur pengujian sistem ini yaitu



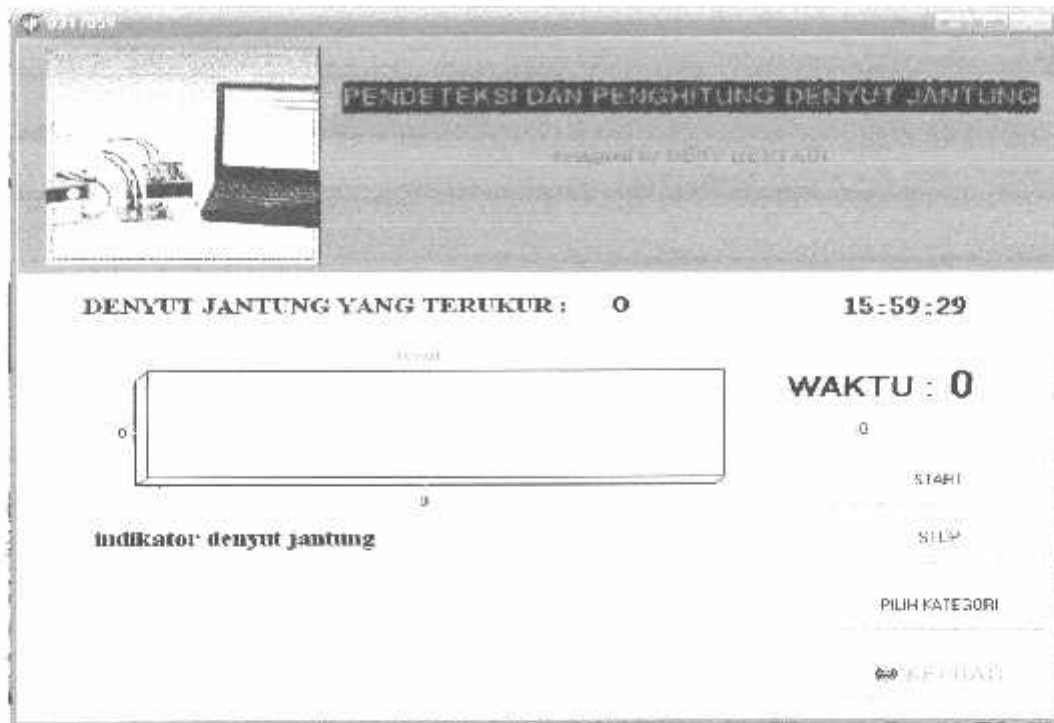
**Gambar 4.19**

#### **Blok Pengujian Sistem Pendeteksi dan Penghitung Denyut Jantung**

- Menghubungkan keseluruhan rangkaian sesuai dengan diagram blok
- Menjalankan program Delphi
- Melakukan proses identifikasi COM
- Memasang tranduser pada tempat pengukuran
- Memulai penghitungan denyut jantung
- Menampilkan hasil pengukuran dan pernyataan hasil pengukuran

❖ Hasil pengujian dari sistem :

1. Tampilan Informasi Pada Program Delphi
  - a. Menampilkan *Form* Utama
  - b. Menampilkan *Form* Waktu, Hasil Pengukuran, Pilih Kategori, Indikator Denyut Jantung, Start, Stop dan Keluar.
  - c. Menampilkan empat pilihan kategori ; bayi baru lahir, anak usia 1-10 tahun, anak usia > 10 tahun + dewasa dan atlet terlatih.
  - d. Menampilkan pernyataan dari jumlah hasil pengukuran ; normal atau tidak normal.



Gambar 4.20

#### Tampilan Awal Menjalankan Program Delphi

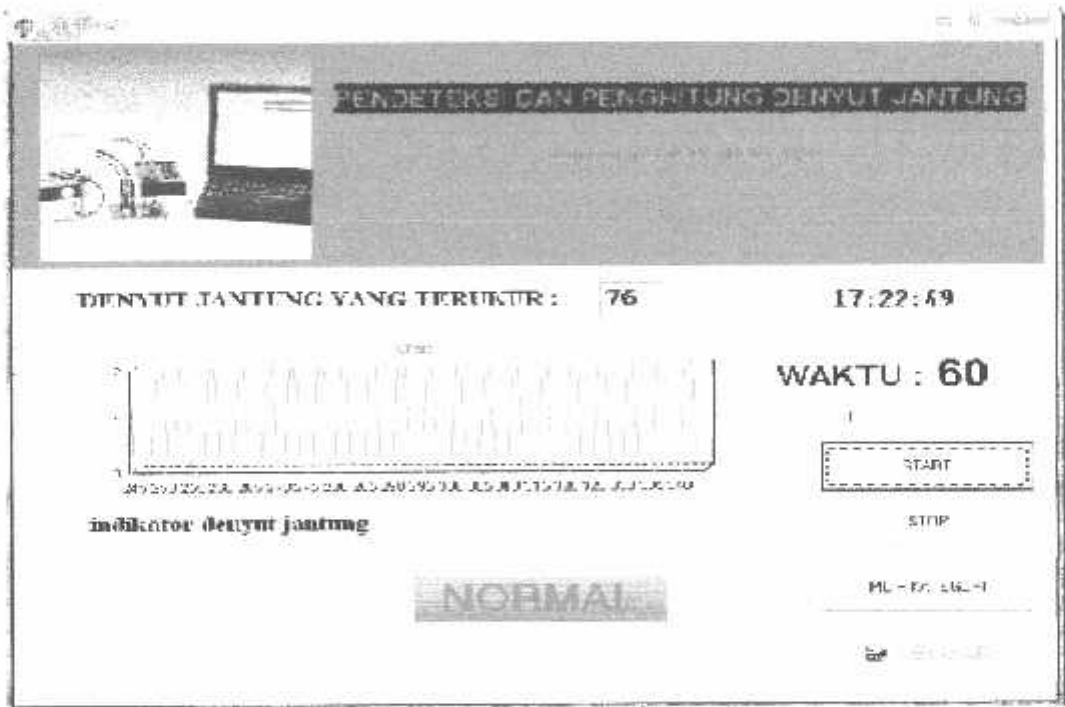
Setelah program dijalankan akan tampil *Form* seperti Gambar 4.15. Kemudian tekan *button* 'pilih kategori' untuk memilih kategori masukan. Maka akan tampil seperti pada Gambar 4.16. Jika transduser telah siap kemudian tekan *button* 'start'. Program mulai melakukan proses penghitungan denyut jantung.





Gambar 4.21

Tampilan Pilihan Kategori pada Program Delphi



Gambar 4.22

Tampilan Hasil Pengukuran dan Pernyataan dari Hasil Pengukuran Setelah Pengukuran Selama 60 Detik pada Program Delphi

Setelah wktu selesai menghitung sampai 60 detik maka program berhenti dan menampilkan pernyataan berdasarkan hasil pengukuran.

Tabel 4.1

Hasil Pengujian Pengukuran Denyut Jantung dengan Tranduser  
Plesthymograf

Pilihan Kategori	Hasil Pengukuran Alat dengan Tranduser Plesthymograf	Hasil Pengukuran BioSys BPM 200	Ket. (Selisih)
Bayi baru lahir		146	-
Anak Usia 1-10 Tahun	97	103	6
Anak Usia >10 Tahun, Dewasa	70	72	2
Atlet Terlatih	74		-

Tabel 4.2

Hasil Pengujian Pengukuran Denyut Jantung dengan Tranduser Akustik

Pilihan Kategori	Hasil Pengukuran dengan Tranduser Akustik	Hasil Pengukuran Manual	Ket. (Selisih)
Bayi baru lahir			-
Anak Usia 1-10 Tahun	83	100	17

Anak Usia > 10 Tahun, Dewasa	64	78	14
Atlet Terlatih	74	75	1

Error yang didapat dari pengujian alat adalah :

$$\text{Error keseluruhan} = \frac{\sum \text{error}}{\sum \text{pengujian}}$$

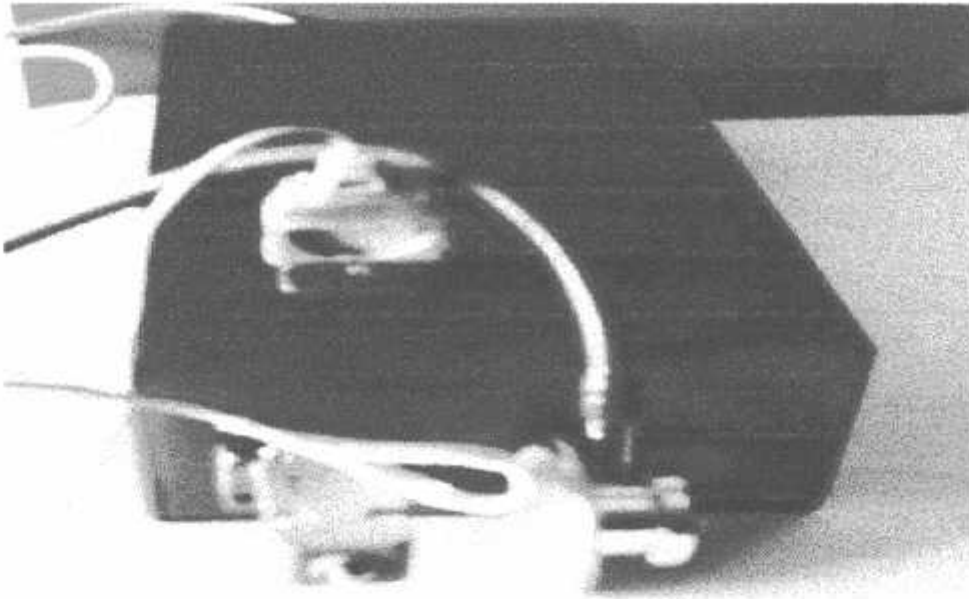
$$\text{Error keseluruhan} = \frac{(5.83 + 2.8 + 17 + 17.9 + 1.3)\%}{5}$$

$$\text{Error keseluruhan} = 8.97 \%$$

#### 4.7. Spesifikasi Alat

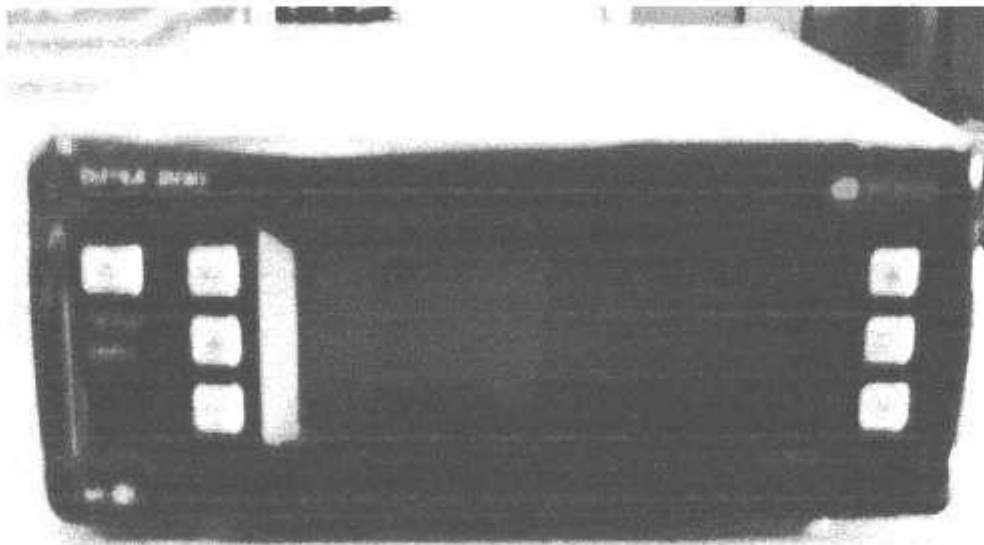
##### Pendeteksi dan Penghitung Denyut Jantung

- Dimensi : 25.5 x 10.5 x 6.5 cm
- Baterai : 10 Volt
- Mikrokontroler AT89C2051
- Transduser Akustik dan Plesthymograf
- Pengkondisi Sinyal
- Modul FT232 USB Serial Converter
- PC
- Perangkat Lunak Pemrograman Delphi



**Gambar 4.23**

**Foto Keseluruhan Alat**



**Gambar 4.24**

**Alat Pembanding pada Proses Pengujian**

**(Pulse Oximeter BioSys BPM 200)**

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Berdasarkan perancangan dan pembuatan pendeteksi dan penghitung denyut jantung manusia berbasis mikrokontroler AT89C2051 ini maka dapat disimpulkan

1. Dari hasil pengujian, pada saat konektor kabel USB tipe A dihubungkan ke PC dengan *port* USB yang berbeda maka PC akan mendeteksi COM yang berbeda.
2. Pada saat melakukan pengujian pengiriman data indikator TX pada modul FT232 USB Serial *Converter* menyala dan pada proses penerimaan data dari mikrokontroler indikator RX menyala.
3. Dari hasil pengujian, dengan empat (4) pilihan kategori terdapat beberapa perbedaan hasil pengukuran dengan pengukuran menggunakan alat *pulse oximeter BioSys BPM 200* di Rumah Sakit Dr. Saiful Anwar Malang ruang IKA serta secara manual. Secara keseluruhan dari hasil pengujian didapat *error* sebesar 8,97%.
4. Dari hasil pengujian, letak dan posisi tranduser sangat mempengaruhi hasil pengukuran. Hal ini dikarenakan denyut tidak terdeteksi oleh tranduser.
5. Karena kondisi orang yang diukur berbeda satu sama lain, misalnya berat badan, warna kulit, ketebalan kulit, ketebalan kuku serta kondisi tempat pengukuran maka perlu adanya tranduser yang peka agar

## 5.2. Saran

Aplikasi pendeteksi dan penghitung denyut jantung manusia berbasis mikrokontroler AT89C2051 ini masih memiliki keterbatasan, nantinya diharapkan dapat dikembangkan untuk mengatasi keterbatasan itu. Seperti untuk membuat transduser yang lebih peka terhadap denyut jantung serta rangkaian pengkondisi sinyal yang menghasilkan sinyal keluaran lebih baik.

---

## DAFTAR PUSTAKA

- [1] Innovative\_electronics
  - [2] Sutawi, Dwi, "I/O BUS & Motherboard" Andi, Yogyakarta, 2004.
  - [3] Data Sheet AT89C2051
  - [4] [www.cvpbiotechnology.com](http://www.cvpbiotechnology.com) , Cardiovascular Physiology Concepts, Richard E. Klabunde, Ph.D, Lippincott Williams & Wilkins, 2005.
  - [5] [www.mystethoscope.com/images/scope\\_anatomy.gif](http://www.mystethoscope.com/images/scope_anatomy.gif)
  - [6] <http://www.kpscc.freeuk.com/symbol.htm>
  - [7] Team # 3, "P.O. Pro *WIRELESS REFLECTANCE PULSE OXIMETER* Design 1, 2004
  - [8] THE MEASUREMENT OF BLOOD PRESSURE IN THE HUMAN BODY, By C. R. Smith and W. H. Bickley C, Clyde Williams and Company, Ohio, 1964
  - [9] [www.nellcor.com](http://www.nellcor.com)
  - [10] [www.gait.edu/~j6-0711/Spring-2006/CIH00694A-A64B-4B4F-A53A-914C1FD3C73D/0/heart\\_rate\\_lab.pdf](http://www.gait.edu/~j6-0711/Spring-2006/CIH00694A-A64B-4B4F-A53A-914C1FD3C73D/0/heart_rate_lab.pdf)
  - [11] [http://en.wikipedia.org/wiki/Heart\\_rate](http://en.wikipedia.org/wiki/Heart_rate)
  - [12] Data Sheet LM358
  - [13] Data Sheet LM311
  - [14] Datasheet LF351
  - [15] BioSys BPM 200 Specifications
-

## *Lampiran*

## *Lampiran*

---





### FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : Deny Heryadi  
NIM : 03 17 059  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Masa Bimbingan : 3 November 2007 s/d 3 Mei 2008  
Judul Skripsi : Perencanaan dan Pembuatan Pendeteksi dan Penghitung Denyut Jantung Manusia Berbasis Mikrokontroler AT89C2051 Dikomunikasikan ke PC Melalui USB

Penguji/Tanggal	Uraian	Paraf
Penguji I 19 Maret 2008	Abstrak	
	Pengujian (Ditambah <i>Error</i> )	
	Kesimpulan	
	Tambahkan Rangkaian Catu Daya DC 5 V	
Penguji II 19 Maret 2008	Abstrak	
	Blok Diagram Disesuaikan dengan Rangkaian	
	Hal. 33 Rangkaian Gambar 3.5 Dilengkapi	
	Hal. 35 rangkaian Pembanding	
	Hal. 37 Rangkaian <i>Buffer</i> ( <i>Converter + Buffer</i> )	
	Hal. Gambar Simbol	

Mengetahui,

Dosen Pembimbing I

( Ir. F. Yudi Limpraptono, MT )  
NIP.Y. 1039500274

Dosen Pembimbing II

( Joseph Dedy Irawan, ST, MT )  
NIP.132315178

Dosen Penguji,

Penguji I

( Ir. Teguh Herbasuki, MT )  
NIP.Y. 1038900209

Penguji II

( Komang Somawirata, ST, MT )  
NIP.Y. 1030100361



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO


### Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : DENY HERYADI  
NIM : 0317  
Perbaikan meliputi :

- Abstrak &
- Pengujian (ditambah error) &
- Kesimpulan &
- Ditambah rangkai catu daya DC 5V &

Malang,

()



### Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : DEAY HERYADI  
NIM : 63.17.059  
Perbaikan meliputi :

- 1) Abstract
- 2) Block Diagram di susunlah dgn rangkai.
- 3) bab 23. Rangkai Gbr 3.5 di tangkas.
- 4) bab 35. Rangkai pemondan.
- 5) bab 37. Rangkai Benflok. → Konverter  
→ Benzon
- 6) bab 39 Gbr Ciri-bul yg benar.

Malang, 17-03-2008

Komang S. Satri



### FORMULIR BIMBINGAN SKRIPSI

Nama : Deny Heryadi  
Nim : 03.17.059  
Masa Bimbingan : 03-November-2007 s/d 03-Mei-2008  
Judul Skripsi : Perencanaan dan Pembuatan Pendeteksi dan Penghitung Denyut Jantung Manusia Berbasis Mikrokontroler AT89C51 Dikomunikasikan ke PC Melalui USB

No	Tanggal	Uraian	Paraf Pembimbing
1	2/2/08	Perbaiki Bab I, II, III	
2	16/2/08	Bab I, II, III	
3		Bab IV	
4		Bab V	
5		Seminar	
6		Demo	
7			
8			
9			
10			

Malang,

Dosen pembimbing I

Ir. F. Yudi Lippapriono, MT  
NIP.Y. 1039500274

Form S-4a



FORMULIR BIMBINGAN SKRIPSI

Nama : Deny Heryadi  
Nim : 03.17.059  
Masa Bimbingan : 03-November-2007 s/d 03-Mei-2008  
Judul Skripsi : Perencanaan dan Pembuatan Pendeteksi dan Penghitung Denyut Jantung Manusia Berbasis Mikrokontroler AT89C2051  
Dikomunikasikan ke PC Melalui USB

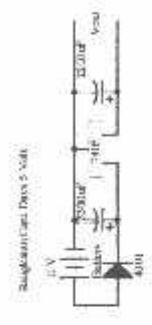
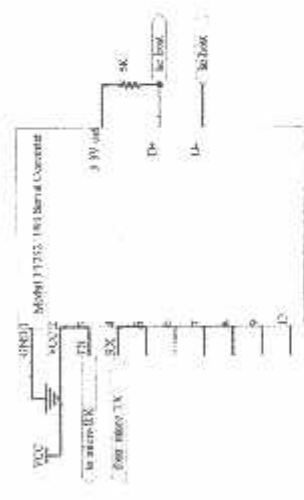
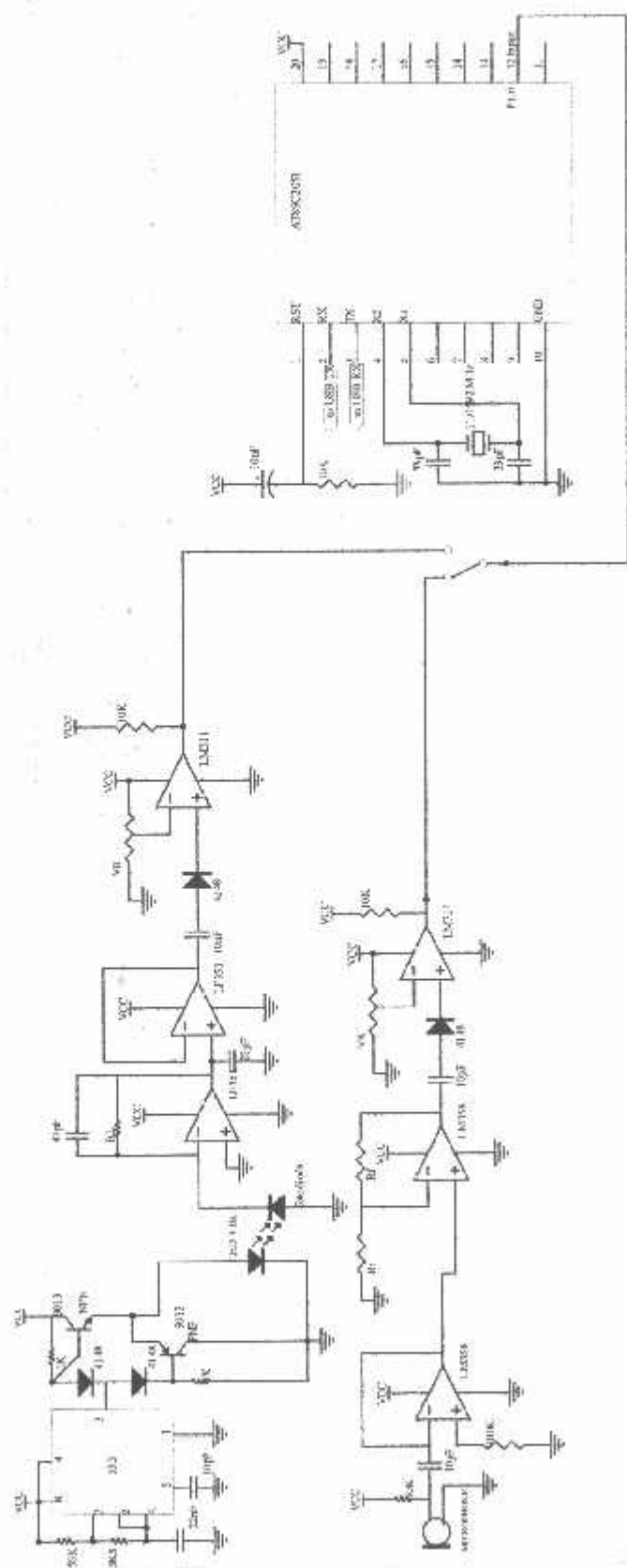
No	Tanggal	Uraian	Paraf Pembimbing
1	03.01. 2008	Landasan Teori, Daftar Pustaka	
2	30.01. 2008	Penguatan Transduser, Sensor	
3	09.02. 2008	Perhitungan, Flowchart	
4	07.02. 2008	Pengujian sensor, alat, Abstraksi	
5	14.02. 2008	Acc Seluruh Hasil	
6	15.02. 2008	Perbaiki Transduser	
7	20.02. 2008	Perbandingan Pengujian Alat	
8	04.03. 2008	ACC Ujian Kompe	
9			
10			

Malang,

Dosen pembimbing II

**Joseph Dedy Irawan, ST, MT**  
NIP.132315178

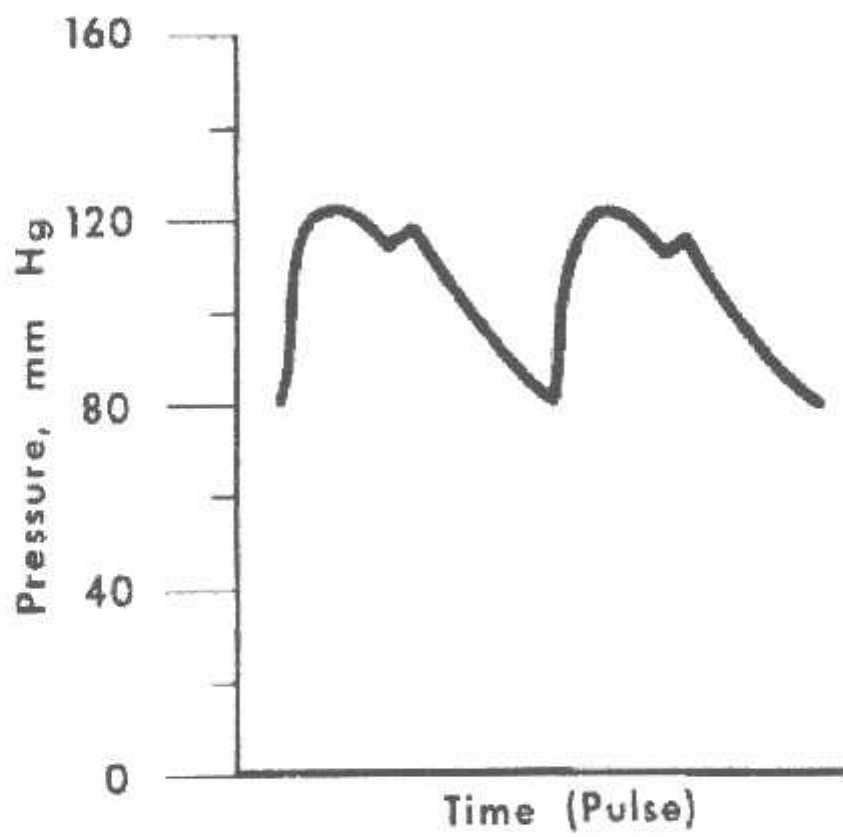
Form S-4b



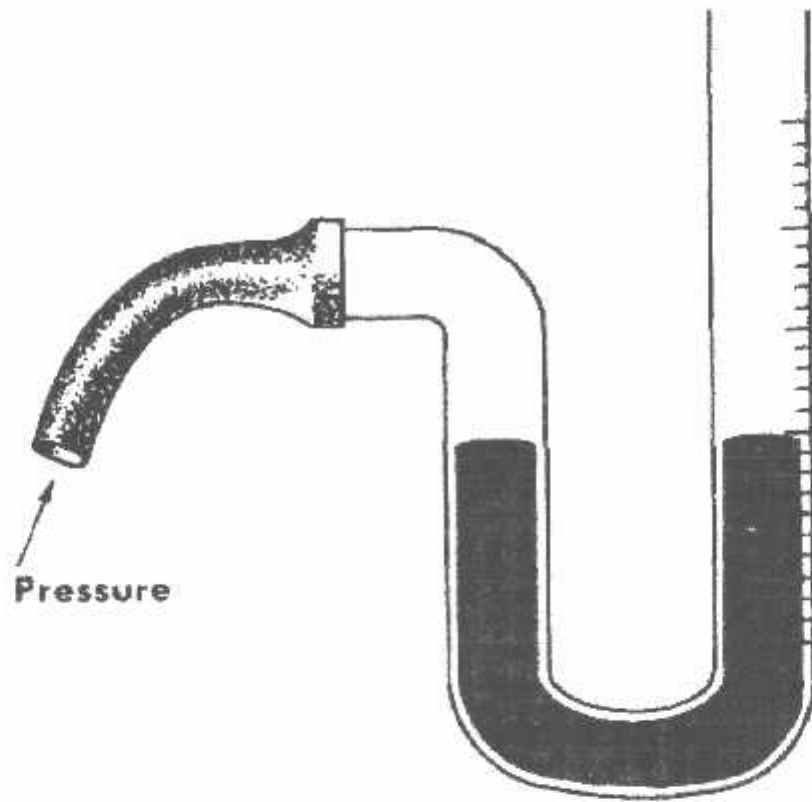
Rev	Rev	Rev	Rev
1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4



Fotodioda



Kurva *Pressure Pulse* pada Jantung Normal

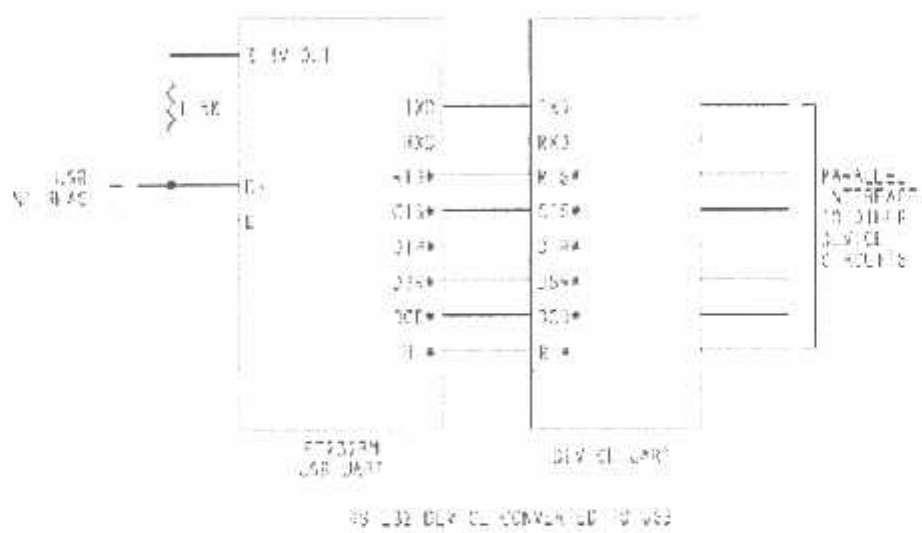
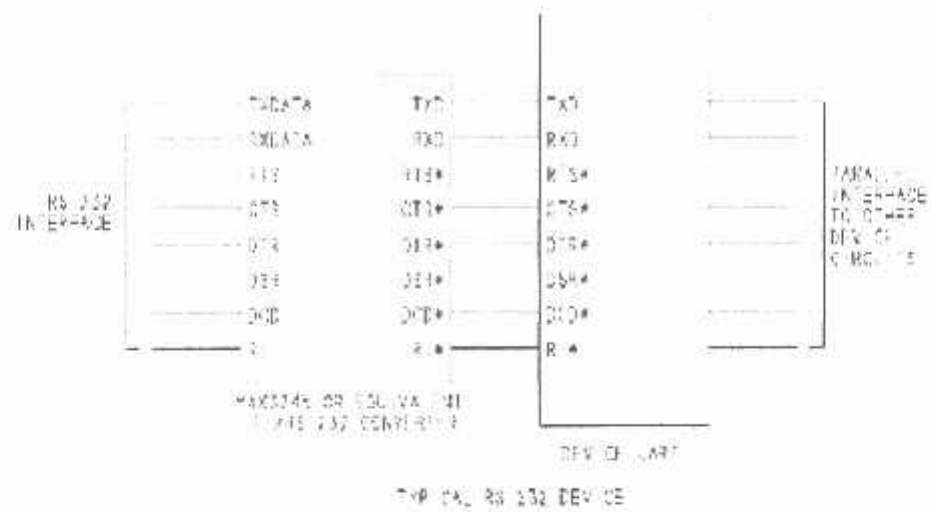


Alat Ukur Tekanan Darah Metode *Invasive* (Konvensional)



Beberapa macam Plesthymograf





### Pengkonversian USB - Serial (RS-232)

# LISTING PROGRAM MIKROKONTROLER AT89C2051

ORG 0000H

JMP MULAI

MULAI:

```
MOV SCON,#50H
MOV TMOD,#20H
MOV TL1,#0FDH
MOV TH1,#0FDH
SETB TR1
SETB EA      ; enable interrupt
SETB ES      ; enable serial interrupt
```

BACA\_DATA:

```
JNB P1.0,KIRIM_DATA_JANTUNG
NOP
JNB P1.0,KIRIM_DATA_JANTUNG
NOP
JNB P1.0,KIRIM_DATA_JANTUNG
NOP
JMP BACA_DATA
```

KIRIM\_DATA\_JANTUNG:

```
MOV B,#31H
CLR ES      ; matikan serial interrupt saat mengirim
MOV A,B
MOV SBUF,A  ; isi serial buffer dengan data yg dikirim
JNB TI,$    ; tunggu pengiriman selesai
CLR TI      ; clear transmit interrupt flag
MOV R5,#0
DJNZ R5,$
SETB ES
MOV R7,#50
OP2: MOV R6,#50
OP1: MOV R5,#50
DJNZ R5,$
DJNZ R6,OP1
DJNZ R7,OP2

JMP BACA_DATA

END
```

## LISTING PROGRAM DELPHI

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, VaClasses, VaComm, StdCtrls, Buttons, ExtCtrls, TeEngine,  
Series, TeeProcs, Chart;
```

```
type
```

```
TForm1 = class(TForm)  
  Panel1: TPanel;  
  Image1: TImage;  
  Label33: TLabel;  
  Label34: TLabel;  
  BitBtn2: TBitBtn;  
  VaComm1: TVaComm;  
  Edit1: TEdit;  
  Label1: TLabel;  
  BitBtn4: TBitBtn;  
  Memo1: TMemo;  
  Timer1: TTimer;  
  Timer2: TTimer;  
  Panel11: TPanel;  
  VaComm2: TVaComm;  
  Edit4: TEdit;  
  Timer3: TTimer;  
  Button2: TButton;  
  Button3: TButton;  
  Timer4: TTimer;  
  Label2: TLabel;  
  Chart1: TChart;  
  Series1: TFastLineSeries;  
  Timer5: TTimer;  
  Edit5: TEdit;  
  Label3: TLabel;  
  GroupBox1: TGroupBox;  
  Button1: TButton;  
  Button4: TButton;  
  Button5: TButton;  
  Button6: TButton;  
  Button7: TButton;  
  Edit2: TEdit;  
  Edit3: TEdit;  
  Panel2: TPanel;  
  Panel3: TPanel;
```

---

```

        Pesan := "";
        form1.Timer2.Enabled := false;
        hitung1 := hitung1 + 1;
        form1.Edit1.Text := inttostr(hitung1);
        dataseri:= form1.Edit1.Text,
        //form1.Timer2.Enabled := true;

    end;
    END;

procedure TForm1.BitBtn4Click(Sender: TObject);
begin
    APPLICATION.Terminate;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    form1.Timer2.Enabled := false;
    form1.Timer3.Enabled := false;
    form1.Timer4.Enabled := false;
    form1.Timer5.Enabled := false;
    form1.Timer6.Enabled := false;
    FORM1.GroupBox1.Hide;
    FORM1.Panel2.Hide;
    FORM1.Panel3.Hide;
    FORM1.Edit2.Hide;
    FORM1.Edit3.Hide;
    FORM1.Panel4.Hide;
    FORM1.Edit5.Hide;
    VaComm1.Close;
    VaComm2.Close;
    VaComm3.Close;
    VaComm4.Close;
    VaComm5.Close;
    VaComm6.Close;
    VaComm7.Close;
    VaComm8.Close;
    VaComm9.Close;
    VaComm10.Close;

    hitung := 0;
    hitung1 := 0;
    //VaComm1.Open;
    FORM1.Memo1.Hide;
end;

procedure TForm1.Timer2Timer(Sender: TObject);
var dataseri : STRING;

```

---

```

begin
form1.Timer2.Enabled := false;
hitung1 := hitung1 + 1;
form1.Edit1.Text := inttostr(hitung1);
dataseri:= form1.Edit1.Text;

END;

procedure TForm1.Timer1Timer(Sender: TObject);
var
    datajam,perwaktu,conter : integer;
    Jam,Menit,Detik,MiliDetik : Word;
    Jam1,menit1,detik1,msec1:word;
    Totalwaktu      :TDateTime;

begin
    DecodeTime(Time, Jam , Menit, Detik, MiliDetik);
    inc (datajam);

    Panel11.Caption := IntToStr(Jam) + ':' +
        IntToStr(Menit) + ':' +
        IntToStr(Detik);
    inc (perwaktu);
end;

procedure TForm1.VaComm2RxChar(Sender: TObject; Count: Integer);
var
    I: Integer;
    Tmp: string;
    dataseri : STRING;
begin
    //baca data yang diterima oleh comport
    Tmp := VaComm1.ReadText;
    form1.Edit5.Text := '';
    for I := 1 to Length(Tmp) do
        case Tmp[I] of
            #10:; //lewatkan
            #13: //Tunggu Enter
            begin
                form1.Memo1.Lines.Add(Pesan);;
                Pesan := ''; //reset Pesan
            end;
            else //bukan #10 atau #13
                Pesan := Pesan + Tmp[I];
                form1.Edit5.Text := Pesan;
                Pesan := '';
                form1.Timer2.Enabled := false;
        end;
    end;

```

---

```

hitung1 := hitung1 + 1;
form1.Edit1.Text := inttostr(hitung1);
dataseri:= form1.Edit1.Text;
    //form1.Timer2.Enabled := true;

    end;
    END;

procedure TForm1.Timer3Timer(Sender: TObject);
begin
    hitung := hitung + 1;
    form1.Edit4.Text := inttostr(hitung);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    FORM1.Pane14.Show;

end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    VaComm1.Close;
    VaComm2.Close;
    VaComm3.Close;
    VaComm4.Close;
    VaComm5.Close;
    VaComm6.Close;
    VaComm7.Close;
    VaComm8.Close;
    VaComm9.Close;
    VaComm10.Close;
    form1.Timer3.Enabled := false;
    form1.Timer4.Enabled := false;
    form1.Timer5.Enabled := false;
end;

procedure TForm1.Timer4Timer(Sender: TObject);
var
    da,di,du : integer;
begin
    //form1.Timer4.Enabled := false;
    if hitung >= 60 then
        begin
            VaComm1.Close;
            VaComm2.Close;
            VaComm3.Close;
            VaComm4.Close;

```

---

```

VaComm5.Close;
VaComm6.Close;
VaComm7.Close;
VaComm8.Close;
VaComm9.Close;
VaComm10.Close;
form1.Timer4.Enabled := false;
form1.Timer3.Enabled := false;
form1.Timer5.Enabled := false;
form1.Timer6.Enabled := true;
end;

end;

procedure TForm1.Timer5Timer(Sender: TObject);
var
    t: longint;
    y: array [1..3] of double;
    V1,V2,V3:STRING;
    input1,input2,input3: real;
    data_jantung: integer;
begin
    data_jantung := strtoint(form1.Edit5.Text);
    if data_jantung >= 1 then
        begin
            x:= x+1;
            y[1]:= 2;
            y[2]:= 3;
            y[3]:= 4;
            v1:= floattostrf(y[1],ffixed,4,0);
            v2:= floattostrf(y[2],ffixed,4,0);
            v3:= floattostrf(y[3],ffixed,4,0);

            form2.memo1.lines.add(inttostr(x)+"V1"+"V2"+"V3");
            //label1.caption:= ' '+v1+ ' mvolt';
            //label2.caption:= ' '+v2+ ' mvolt';
            //label3.caption:= ' '+v3+ ' mvolt';
            for t:=0 to chart1.SeriesCount-1 do
                with chart1.Series[t] do
                    add(y[t-1],clteecolor);
                    with chart1.BottomAxis do
                        begin
                            automatic := false;
                            maximum := series1.XValues.Last;
                            minimum:= maximum-100;
                        end;
                    end;

            x:= x+1;

```

```

y[1]:= 2;
y[2]:= 3;
y[3]:= 4;
v1:= floattostrf(y[1],ffixed,4,0);
v2:= floattostrf(y[2],ffixed,4,0);
v3:= floattostrf(y[3],ffixed,4,0);

form2.memo1.lines.add(inttostr(x)+"V1"+"V2"+"V3);
//label1.caption:=' '+v1+ ' mvolt';
//label2.caption:=' '+v2+ ' mvolt';
//label3.caption:=' '+v3+ ' mvolt';
for t:=0 to chart1.SeriesCount-1 do
  with chart1.Series[t] do
    add(y[t+1],"clteecolor);
    with chart1.BottomAxis do
      begin
        automatic := false;
        maximum := series1.XValues.Last;
        minimum:= maximum-100;
      end;

```

```

form1.Edit5.Text := '0';
x:= x+1;
y[1]:= 0;
y[2]:= 3;
y[3]:= 4;
v1:= floattostrf(y[1],ffixed,4,0);
v2:= floattostrf(y[2],ffixed,4,0);
v3:= floattostrf(y[3],ffixed,4,0);

```

```

form2.memo1.lines.add(inttostr(x)+"V1"+"V2"+"V3);
//label1.caption:=' '+v1+ ' mvolt';
//label2.caption:=' '+v2+ ' mvolt';
//label3.caption:=' '+v3+ ' mvolt';
for t:=0 to chart1.SeriesCount-1 do
  with chart1.Series[t] do
    add(y[t+1],"clteecolor);
    with chart1.BottomAxis do
      begin
        automatic := false;
        maximum := series1.XValues.Last;
        minimum:= maximum-100;
      end;
    end
  else
    begin
      x:= x+1;
      y[1]:= 0;

```

---



```

y[2]:= 3;
y[3]:= 4;
v1:= floattostrf(y[1],ffixed,4,0);
v2:= floattostrf(y[2],ffixed,4,0);
v3:= floattostrf(y[3],ffixed,4,0);

form2.memo1.lines.add(inttostr(x)+"V1"+"V2"+"V3);
    //label1.caption:=' '+v1+ ' mvolt';
    //label2.caption:=' '+v2+ ' mvolt';
    //label3.caption:=' '+v3+ ' mvolt';
    for t:=0 to chart1.SeriesCount-1 do
        with chart1.Series[t] do
add(y[t+1],",",clteecolor);
        with chart1.BottomAxis do
            begin
                automatic := false;
                maximum := series1.XValues.Last;
                minimum:= maximum-100;
            end;
        end;
    end;

procedure TForm1.Button7Click(Sender: TObject);
begin
FORM1.GroupBox1.Show;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
FORM1.Edit2.Text := '119';
FORM1.Edit3.Text := '161';
FORM1.GroupBox1.Hide;
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
FORM1.Edit2.Text := '79';
FORM1.Edit3.Text := '121';
FORM1.GroupBox1.Hide;
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
FORM1.Edit2.Text := '59';
FORM1.Edit3.Text := '81';
FORM1.GroupBox1.Hide;
end;

```

---

```

procedure TForm1.Button6Click(Sender: TObject);
begin
FORM1.Edit2.Text := '39';
FORM1.Edit3.Text := '61';
FORM1.GroupBox1.Hide;
end;

procedure TForm1.Panel2Click(Sender: TObject);
begin
FORM1.Panel2.Hide;
end;

procedure TForm1.Panel3Click(Sender: TObject);
begin
FORM1.Panel3.Hide;
end;

procedure TForm1.Timer6Timer(Sender: TObject);
var
da,di,du : integer;
begin
form1.Timer6.Enabled := false;
da := strtoint(form1.Edit1.Text);
di := strtoint(form1.Edit2.Text);
du := strtoint(form1.Edit3.Text);
FORM1.Panel2.Hide;
FORM1.Panel3.Hide;
if da <= di then
begin
FORM1.Panel3.Show;
end;
if da >= du then
begin
FORM1.Panel3.Show;
end;
if da >= di then
begin
if da <= du then
begin
FORM1.Panel2.Show;
end
end
end;
end;
procedure TForm1.Button9Click(Sender: TObject);
begin
FORM1.Panel4.Hide;
FORM1.VaComm1.Open;
hitung := 0;

```

---

```
hitung1 := 0;  
form1.Timer3.Enabled := true;  
form1.Timer4.Enabled := true;  
form1.Timer5.Enabled := TRUE;  
end;
```

```
procedure TForm1.Button10Click(Sender: TObject);  
begin  
FORM1.Panel4.Hide;  
FORM1.VaComm2.Open;  
hitung := 0;  
hitung1 := 0;  
form1.Timer3.Enabled := true;  
form1.Timer4.Enabled := true;  
form1.Timer5.Enabled := TRUE;  
end;
```

```
procedure TForm1.Button11Click(Sender: TObject);  
begin  
FORM1.Panel4.Hide;  
FORM1.VaComm3.Open;  
hitung := 0;  
hitung1 := 0;  
form1.Timer3.Enabled := true;  
form1.Timer4.Enabled := true;  
form1.Timer5.Enabled := TRUE;  
end;
```

```
procedure TForm1.Button12Click(Sender: TObject);  
begin  
FORM1.Panel4.Hide;  
FORM1.VaComm4.Open;  
hitung := 0;  
hitung1 := 0;  
form1.Timer3.Enabled := true;  
form1.Timer4.Enabled := true;  
form1.Timer5.Enabled := TRUE;  
end;
```

```
procedure TForm1.Button13Click(Sender: TObject);  
begin  
FORM1.Panel4.Hide;  
FORM1.VaComm5.Open;  
hitung := 0;  
hitung1 := 0;  
form1.Timer3.Enabled := true;  
form1.Timer4.Enabled := true;  
form1.Timer5.Enabled := TRUE;
```

---

end;

```
procedure TForm1.Button14Click(Sender: TObject);
begin
  FORM1.Panel4.Hide;
  FORM1.VaComm6.Open;
  hitung := 0;
  hitung1 := 0;
  form1.Timer3.Enabled := true;
  form1.Timer4.Enabled := true;
  form1.Timer5.Enabled := TRUE;
end;
```

```
procedure TForm1.Button15Click(Sender: TObject);
begin
  FORM1.Panel4.Hide;
  FORM1.VaComm7.Open;
  hitung := 0;
  hitung1 := 0;
  form1.Timer3.Enabled := true;
  form1.Timer4.Enabled := true;
  form1.Timer5.Enabled := TRUE;
end;
```

```
procedure TForm1.Button16Click(Sender: TObject);
begin
  FORM1.Panel4.Hide;
  FORM1.VaComm8.Open;
  hitung := 0;
  hitung1 := 0;
  form1.Timer3.Enabled := true;
  form1.Timer4.Enabled := true;
  form1.Timer5.Enabled := TRUE;
end;
```

```
procedure TForm1.Button17Click(Sender: TObject);
begin
  FORM1.Panel4.Hide;
  FORM1.VaComm9.Open;
  hitung := 0;
  hitung1 := 0;
  form1.Timer3.Enabled := true;
  form1.Timer4.Enabled := true;
  form1.Timer5.Enabled := TRUE;
end;
```

```
procedure TForm1.Button18Click(Sender: TObject);
begin
```

---

```

FORM1.Panel4.Hide;
FORM1.VaComm10.Open;
hitung := 0;
hitung1 := 0;
form1.Timer3.Enabled := true;
form1.Timer4.Enabled := true;
form1.Timer5.Enabled := TRUE;
end;

```

```

procedure TForm1.VaComm3RxChar(Sender: TObject; Count: Integer);
var
  I: Integer;
  Tmp: string;
  dataseri : STRING;
begin
  //baca data yang diterima oleh comport
  Tmp := VaComm3.ReadText;
  form1.Edit5.Text := '';
  for I := 1 to Length(Tmp) do
    case Tmp[I] of
      #10; //lewatkan
      #13; //Tunggu Enter
      begin
        form1.Memo1.Lines.Add(Pesan);
        Pesan := ''; //reset Pesan
      end;
      else //bukan #10 atau #13
        Pesan := Pesan + Tmp[I];
        form1.Edit5.Text := Pesan;
        Pesan := '';
        form1.Timer2.Enabled := false;
    end;
  hitung1 := hitung1 + 1;
  form1.Edit1.Text := inttostr(hitung1);
  dataseri := form1.Edit1.Text;
  //form1.Timer2.Enabled := true;

  end;
END;

```

```

procedure TForm1.VaComm4RxChar(Sender: TObject; Count: Integer);
var
  I: Integer;
  Tmp: string;
  dataseri : STRING;
begin
  //baca data yang diterima oleh comport
  Tmp := VaComm4.ReadText;
  form1.Edit5.Text := '';

```

---

```

for I := 1 to Length(Tmp) do
  case Tmp[I] of
    #10:; //lewatkan
    #13: //Tunggu Enter
      begin
        form1.Memo1.Lines.Add(Pesan);
        Pesan := ''; //reset Pesan
      end;
    else //bukan #10 atau #13
      Pesan := Pesan + Tmp[I];
      form1.Edit5.Text := Pesan;
      Pesan := '';
      form1.Timer2.Enabled := false;
  hitung1 := hitung1 + 1;
  form1.Edit1.Text := inttostr(hitung1);
  dataseri:= form1.Edit1.Text;
  //form1.Timer2.Enabled := true;

end;
END;

```

```

procedure TForm1.VaComm5RxChar(Sender: TObject; Count: Integer);
var
  I: Integer;
  Tmp: string;
  dataseri : STRING;
begin
  //baca data yang diterima oleh comport
  Tmp := VaComm5.ReadText;
  form1.Edit5.Text := '';
  for I := 1 to Length(Tmp) do
    case Tmp[I] of
      #10:; //lewatkan
      #13: //Tunggu Enter
        begin
          form1.Memo1.Lines.Add(Pesan);
          Pesan := ''; //reset Pesan
        end;
      else //bukan #10 atau #13
        Pesan := Pesan + Tmp[I];
        form1.Edit5.Text := Pesan;
        Pesan := '';
        form1.Timer2.Enabled := false;
    end;
  hitung1 := hitung1 + 1;
  form1.Edit1.Text := inttostr(hitung1);
  dataseri:= form1.Edit1.Text;
  //form1.Timer2.Enabled := true;

```

---

```
end;  
END;
```

```
procedure TForm1.VaComm6RxChar(Sender: TObject; Count: Integer);  
var  
  I: Integer;  
  Tmp: string;  
  dataseri : STRING;  
begin  
  //baca data yang diterima oleh comport  
  Tmp := VaComm6.ReadText;  
  form1.Edit5.Text := '';  
  for I := 1 to Length(Tmp) do  
    case Tmp[I] of  
      #10:; //lewatkan  
      #13: //Tunggu Enter  
        begin  
          form1.Memo1.Lines.Add(Pesan);  
          Pesan := ''; //reset Pesan  
        end;  
      else //bukan #10 atau #13  
        Pesan := Pesan + Tmp[I];  
        form1.Edit5.Text := Pesan;  
        Pesan := '';  
        form1.Timer2.Enabled := false;  
    end;  
  hitung1 := hitung1 + 1;  
  form1.Edit1.Text := inttostr(hitung1);  
  dataseri:= form1.Edit1.Text;  
  //form1.Timer2.Enabled := true;  
  
end;  
END;
```

```
procedure TForm1.VaComm7RxChar(Sender: TObject; Count: Integer);  
var  
  I: Integer;  
  Tmp: string;  
  dataseri : STRING;  
begin  
  //baca data yang diterima oleh comport  
  Tmp := VaComm7.ReadText;  
  form1.Edit5.Text := '';  
  for I := 1 to Length(Tmp) do  
    case Tmp[I] of  
      #10:; //lewatkan  
      #13: //Tunggu Enter  
        begin  
          form1.Memo1.Lines.Add(Pesan);
```

---

```

        Pesan := ""; //reset Pesan
    end;
else //bukan #10 atau #13
    Pesan := Pesan + Tmp[1];
    form1.Edit5.Text := Pesan;
    Pesan := "";
    form1.Timer2.Enabled := false;
hitung1 := hitung1 + 1;
form1.Edit1.Text := inttostr(hitung1);
dataseri:= form1.Edit1.Text;
    //form1.Timer2.Enabled := true;

```

```

end;
END;

```

```

procedure TForm1.VaComm8RxChar(Sender: TObject; Count: Integer);
var

```

```

    I: Integer;
    Tmp: string;
    dataseri : STRING;

```

```

begin

```

```

    //baca data yang diterima oleh comport

```

```

    Tmp := VaComm8.ReadText;

```

```

    form1.Edit5.Text := '';

```

```

    for I := 1 to Length(Tmp) do

```

```

        case Tmp[I] of

```

```

            #10; //lewatkan

```

```

            #13; //Tunggu Enter

```

```

        begin

```

```

            form1.Memo1.Lines.Add(Pesan);

```

```

            Pesan := ""; //reset Pesan

```

```

        end;

```

```

    else //bukan #10 atau #13

```

```

        Pesan := Pesan + Tmp[I];

```

```

        form1.Edit5.Text := Pesan;

```

```

        Pesan := "";

```

```

        form1.Timer2.Enabled := false;

```

```

    hitung1 := hitung1 + 1;

```

```

    form1.Edit1.Text := inttostr(hitung1);

```

```

    dataseri:= form1.Edit1.Text;

```

```

    //form1.Timer2.Enabled := true;

```

```

end;

```

```

END;

```

```

procedure TForm1.VaComm9RxChar(Sender: TObject; Count: Integer);
var

```

```

    I: Integer;

```



```

    form1.Timer2.Enabled := false;
    hitung1 := hitung1 + 1;
    form1.Edit1.Text := inttostr(hitung1);
    dataseri := form1.Edit1.Text;
    //form1.Timer2.Enabled := true;

```

```

end;
END;

```

```

procedure TForm1.VaComm2Break(Sender: TObject);
var

```

```

    I: Integer;
    Tmp: string;
    dataseri : STRING;
begin
    //baca data yang diterima oleh comport
    Tmp := VaComm2.ReadText;
    form1.Edit5.Text := '';
    for I := 1 to Length(Tmp) do
        case Tmp[I] of
            #10: //lewatkan
            #13: //Tunggu Enter
            begin
                form1.Memo1.Lines.Add(Pesan);
                Pesan := ''; //reset Pesan
            end;
            else //bukan #10 atau #13
                Pesan := Pesan + Tmp[I];
                form1.Edit5.Text := Pesan;
                Pesan := '';
                form1.Timer2.Enabled := false;
        end;
    end;
    hitung1 := hitung1 + 1;
    form1.Edit1.Text := inttostr(hitung1);
    dataseri := form1.Edit1.Text;
    //form1.Timer2.Enabled := true;

```

```

end;
END;

```

```

end.

```

## Features

- Compatible with MCS-51™ Products
- 2 Kbytes of Reprogrammable Flash Memory  
Endurance: 1,000 Write/Erase Cycles
- 2.7 V to 6 V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes

## Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2 Kbytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2 Kbytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

## in Configuration

PDIP/SOIC

RST	1	20	VCC
PXD/P3.0	2	19	P1.7
TXD/P3.1	3	18	P1.6
XTAL2	4	17	P1.5
XTAL1	5	16	P1.4
INT0/P3.2	6	15	P1.3
INT1/P3.3	7	14	P1.2
XT0/P3.4	8	13	P1.1 (AIN1)
XT1/P3.5	9	12	P1.0 (AIN0)
GND	10	11	P3.7

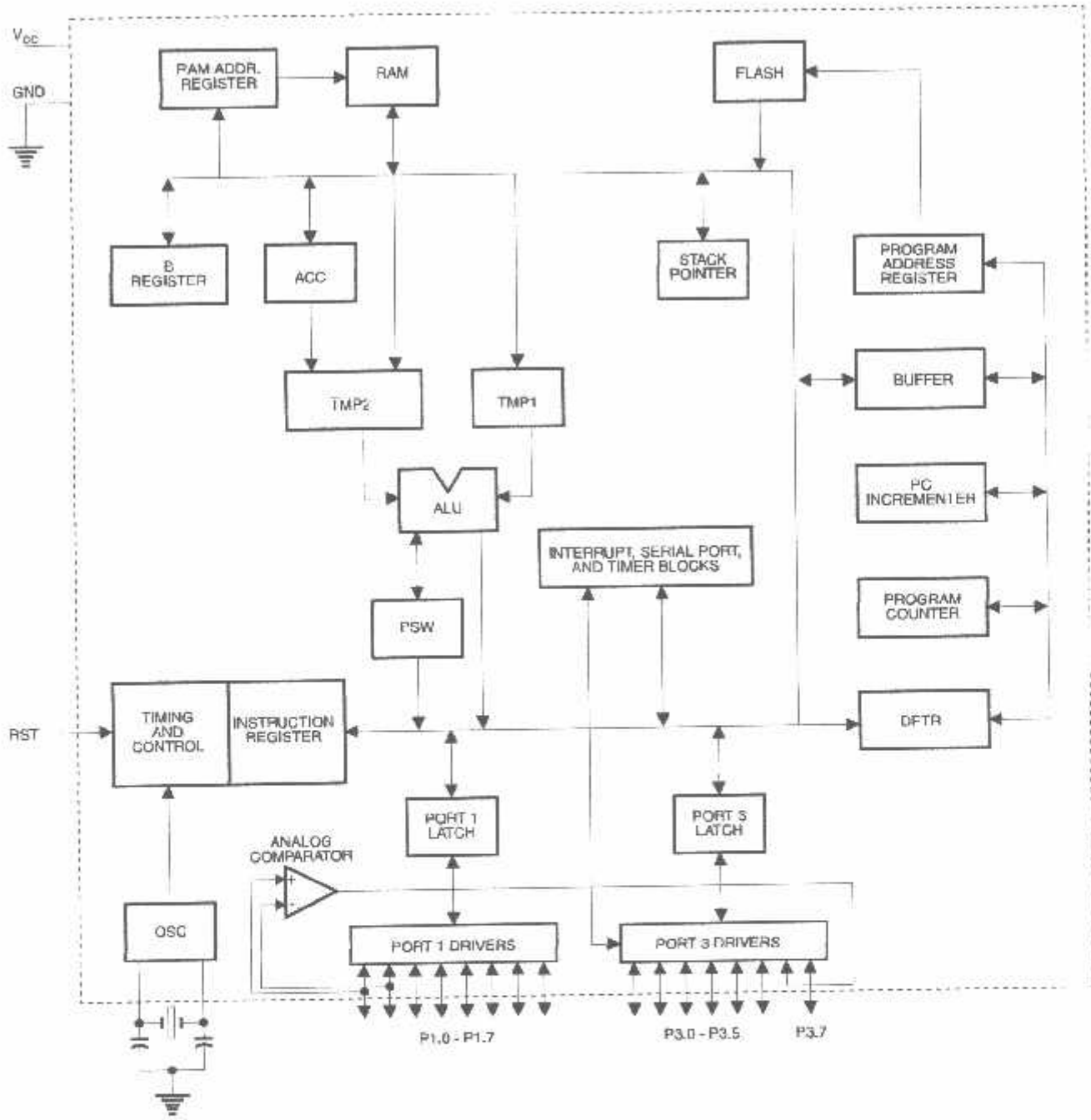
## 8-Bit Microcontroller with 2 Kbytes Flash

0365C

3-17



# Block Diagram



## Pin Description

**V<sub>CC</sub>**  
Supply voltage.

**GND**  
Ground.

**Port 1**  
Port 1 is an 8-bit bidirectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (I<sub>IL</sub>) because of the internal pullups.

Port 1 also receives code data during Flash programming and program verification.

**Port 3**  
Port 3 pins P3.0 to P3.5, P3.7 are seven bidirectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as outputs. As inputs, Port 3 pins that are externally being pulled low will source current (I<sub>IL</sub>) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C2051 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

Port 3 also receives some control signals for Flash programming and programming verification.

**RST**  
Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

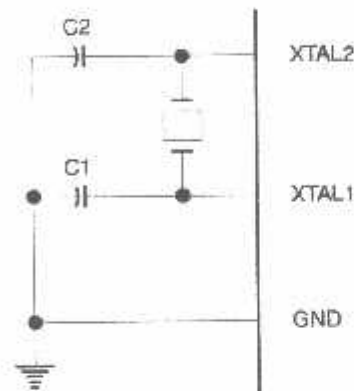
**XTAL1**  
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

**XTAL2**  
Output from the inverting oscillator amplifier.

## Oscillator Characteristics

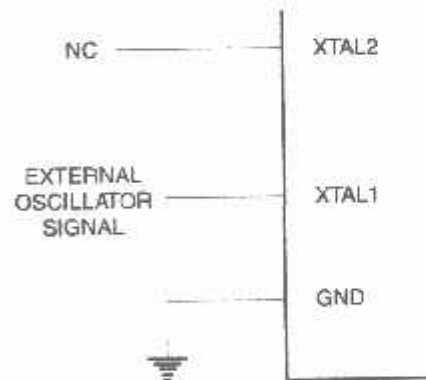
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



Notes: C1, C2 = 30 pF ± 10 pF for Crystals  
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration





## Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in the table below.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return

random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Table 1. AT89C2051 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H								0CFH
0C0H								0C7H
0B8H	IP XXX00000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0XX00000							0AFH
0A0H								0A7H
098H	SCON 00000000	SBUF XXXXXXXX						9FH
090H	P1 11111111							97H
088H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
080H		SP 00000111	DPL 00000000	DPH 00000000			PCON 0XXX0000	87H

**Data Polling:** The AT89C2051 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

**Ready/Busy:** The Progress of byte programming can also be monitored by the RDY/BSY output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

**Program Verify:** If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from 'L' to 'H'.
  2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
  3. Pulse pin XTAL1 once to advance the internal address counter.
  4. Read the next code data byte at the port P1 pins.
  5. Repeat steps 3 and 4 until the entire array is read.
- The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

**Chip Erase:** The entire PEROM array (2 Kbytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

**Reading the Signature Bytes:** The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.





(000H) = 1EH indicates manufactured by Atmel  
(001H) = 21H indicates 89C2051

## Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

## Flash Programming Modes

Mode		RST	P3.2/ PROG	P3.3	P3.4	P3.5	P3.7
Write Code Data <sup>(1,3)</sup>		12V		L	H	H	H
Read Code Data <sup>(1)</sup>		H	H	L	L	H	H
Write Lock	Bit - 1	12V		H	H	H	H
	Bit - 2	12V		H	H	L	L
Chip Erase		12V	 <sup>(2)</sup>	H	L	L	L
Read Signature Byte		H	H	L	L	L	L

Notes: 1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL1 pin.

2. Chip Erase requires a 10 ms PROG pulse.

3. P3.1 is pulled Low during programming to indicate RDY/BSY.



# LF351

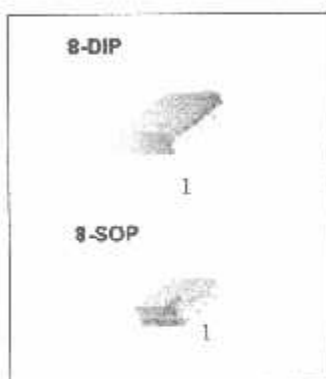
## Single Operational Amplifier (JFET)

### Features

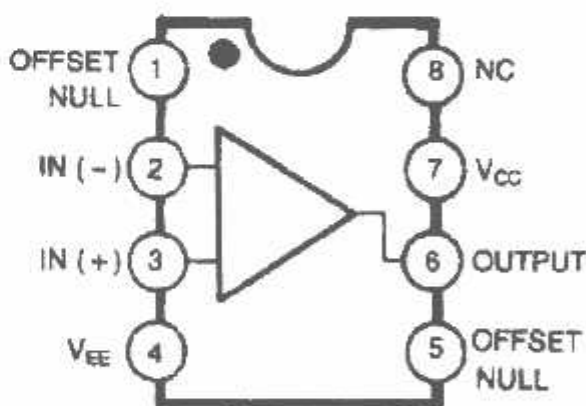
- Internally trimmed offset voltage: 10mV
- Low input bias current : 50pA
- Wide gain bandwidth : 4MHz
- High slew rate : 13V/ $\mu$ s
- High input impedance :  $10^{12}\Omega$

### Description

The LF351 is JFET input operational amplifier with an internally compensated input offset voltage. The JFET input device provides wide bandwidth, low input bias currents and offset currents.

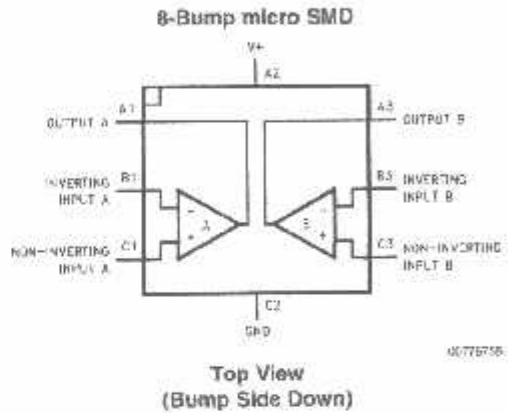
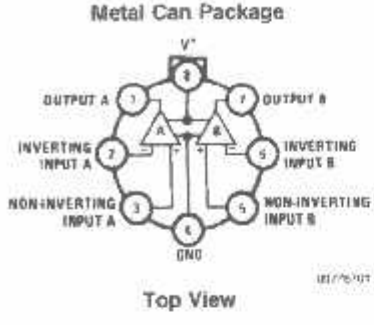
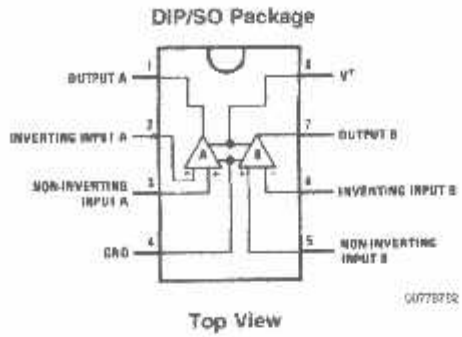


### Internal Block Diagram

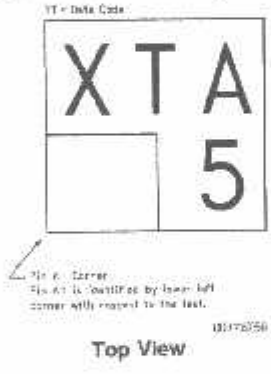


Rev. 1.0.1

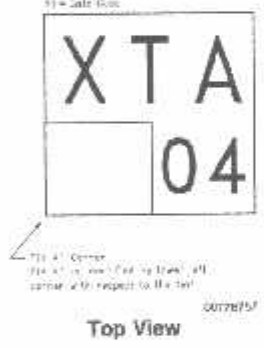
Connection Diagrams



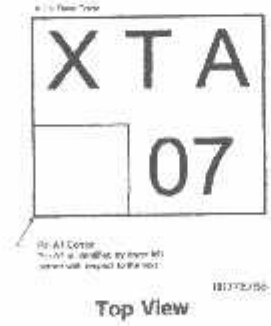
LM358BP micro SMD Marking Orientation



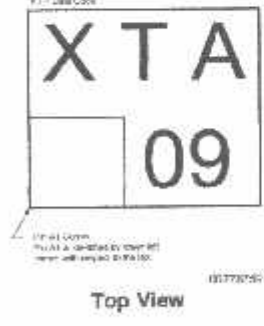
LM2904BP micro SMD Marking Orientation



LM358TP micro SMD Marking Orientation



LM2904TP micro SMD Marking Orientation





# LM311

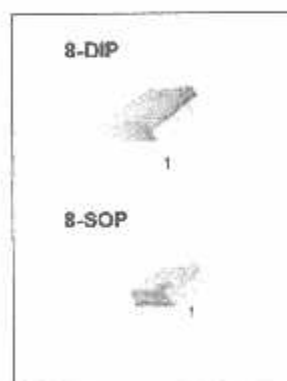
## Single Comparator

### Features

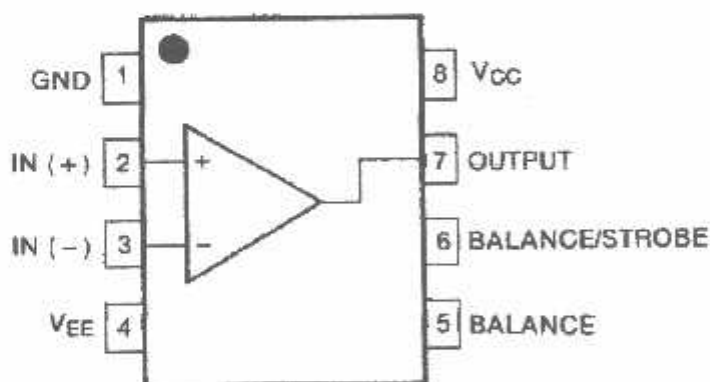
- Low input bias current : 250nA (Max)
- Low input offset current : 50nA (Max)
- Differential Input Voltage :  $\pm 30V$
- Power supply voltage : single 5.0V supply to  $\pm 15V$
- Offset voltage null capability
- Strobe capability

### Description

The LM311 series is a monolithic, low input current voltage comparator. The device is also designed to operate from dual or single supply voltage.



### Internal Block Diagram



Rev. 1.0.1